Abstract

Title: INVESTIGATING THE APPLICATION OF INTERPRETABILITY TECHNIQUES TO COMPUTATIONAL TOXICOLOGY

Aranya Banerjee, Kevin Boby, Samuel Lam, David Polefrone, Robert San, Erika Schlunk, Sean Wynn, Colin Yancey

Directed by: Dr. Soheil Feizi
Department of Computer Science, University of Maryland

A barrier to the incorporation of predictive models for drug design lies in their lack of interpretability. To this end, we examine on three fronts the interpretability of benchmark models for the 2014 Tox21 Data Challenge, an initiative in the domain with a dataset of measurements across twelve toxicity experiments. On existing measures of model performance, we assess the current benchmark metrics' ability to describe model behavior and recommend an alternative set of metrics for the task. On the existing interpretability methods for machine learning models, we quantitatively and qualitatively evaluate their application to this domain by measuring desirable properties of explanations they produce. Additionally, we incorporate a recently described method for partial charge prediction as novel input for a toxicological model and observe its resulting model performance and model interpretability.

Investigating the Application of Interpretability Techniques to Computational Toxicology

by

Team TOXIC

Aranya Banerjee, Kevin Boby, Samuel Lam, David Polefrone, Robert San, Erika Schlunk, Sean Wynn, Colin Yancey

Thesis submitted in partial fulfillment of the requirements of the Gemstone
Honors Program, University of Maryland
2021

Mentor: Dr. Soheil Feizi

Discussants:
Dr. Scott S. Auerbach
Dr. Pratyush Tiwary
Dr. Todd Martin
Dr. Kenneth Frauwirth
Dr. Philippe Youkharibache

Contents

Acknowledgements

We would like to thank our mentor, Dr. Soheil Feizi, for all the invaluable advice he has given our team in the past two years. We would also like to thank Dr. Aravind Srinivasan, our first mentor, who started our team off in a fruitful direction. Without the coordination, structure, and support of the Gemstone staff and program, including Dr. Skendall, Dr. Coale, and Dr. Lovell, none of this would have been possible. Our librarian, Ms. Lindsay Inge, also provided valuable assistance. We would like to thank former team members for all their contributions during their time with us. Finally, we give our warmest regards to our friends and family who supported us all throughout our time in college.

1 Introduction

Toxicology is a critically important scientific field, combining innovations in biology, chemistry, and pharmacology. Toxicologists work in many capacities, including in academics, industry, and regulatory agencies. They may treat cases of poisoning in emergency departments, contribute to pharmaceutical research relating to drug safety, or advise health policy. They study a wide range of chemical compounds that have effects on humans, animals, and the environment. Since toxicology directly relates to the effects of these chemicals on biological life, the primary method of study has been experimentation on biological subjects and tissues. However, a developing area within toxicology, drawing from the availability and power of computational resources, is computational toxicology. The origins of this discipline may be traced to the advent of systematic toxicology investigations conduced by the Environmental Protection Agency (EPA) in the mid-1980s, with a focus on potential exposure hazards of consumer products and other chemicals [1].

Computational toxicology develops models that predict adverse health effects that do not rely on in vivo and in vitro testing and their associated costs [2]. While it does not have the capability of totally replacing these methods, its application could potentially reduce costs, increase speed and efficiency, and provide more ethical approaches to drug testing [3]. Pharmaceutical companies already use computational methods to screen large numbers of compounds before developing the promising candidates further, already demonstrating some of the powerful cost-saving capabilities of computers in today's research [2]. Predictive models hold the potential to reduce the risk of toxic drugs entering the market, in part by reducing wasteful investigations into drugs which might turn out to be toxic later on in trials. One important area of application of toxicology has not yet readily accepted the use of computational models: the regulatory field.

1.1 Drug Approval

In the United States, the Food and Drug Administration (FDA) oversees the drug approval process. Bringing drugs to market with FDA approval occurs in five broad steps. Step 1 involves Drug Discovery and Development, which is the job of pharmaceutical chemists, biologists, and medical experts. Step 2 involves Preclinical Research, which takes place in lab research settings by industrial researchers, who may utilize in vivo and/or in vitro trials. Good Laboratory Practices, as defined by FDA in 21 C.F.R. Part 58.1 [4], guide research in this step.

Step 3 is Clinical Research, which is perhaps the most central in the process. This step involves testing on human cohorts, and occurs in several phases, with increased study sizes at each phase. Should a drug fail any phase, it does not proceed on to subsequent phases. Industrial researchers may consult the FDA for assistance in designing trials. In Phase 1, drugs are tested for their safety and dosage. In Phase 2, drugs are tested for their efficacy and side effects. In Phase 3, drugs are tested for their efficacy, and monitored for adverse reactions. In Phase 4, drugs are tested for their overall safety and efficacy.

If a drug passes each of these phases, it moves on to Step 4, or FDA drug review. This step involves the submission of all previous data to FDA in a New Drug Application (NDA). The reviewers then scrutinize the data for 6 to 10 months and may approve it given a favorable evaluation. Subsequently, FDA and drug developers then work together on appropriate prescribing and labeling. At this point, the drug may arrive at markets, but is not done with the regulatory process. In Step 5, the FDA conducts post-market drug safety monitoring. This step includes various methods of drug monitoring and reevaluating.

1.2 Issues with the Drug Approval Process

The FDA approval process, which is upheld as a global standard, has several critical shortcomings. First, it operates on a slow timetable: the average time for drugs to reach market from their first time of experimentation is 12 years [5]. Additionally, it is financially costly and risky. According to a landmark study by the Tufts Center for the Study of Drug Development, the average total cost to bring a drug to the pre-approval stage is $2.558 billion, in 2013 dollars [6]. Not considering opportunity costs incurred by delays in the process, the average out-of-pocket financial cost is $1.395 billion, in 2013 dollars [6]. The levels of safety assurance it provides are also arguably not efficient, given the lack of context dependence involved. Even if the statistical thresholds for testing false positives and false negatives are appropriately slim on the aggregate, they may not be appropriate as a dynamic standard from condition to condition. In some life-or-death cases, false negatives may be much more life-threatening than false positives. The converse applies to routine, low-level conditions, which have small adverse effects across a large population, since there is then a larger potential hazard promulgated across a much wider group of individuals [7].

Finally, and perhaps most notably, the process itself carries some degree of danger. In 1993, five patients died in a Phase I trial of fialuridine, a seemingly exciting new drug for the treatment of hepatitis B [8]. In 2006, six patients experienced multi-organ failure in a Phase I trial of TGN1412, an antibody intended to fight autoimmune disorders. Researchers had given the antibody to subjects at a dose 500 times lower than that which was deemed safe after animal trials [9]. This is not intended to be a completely pessimistic viewpoint of FDA's approval process, which is built under well-meaning principles and certainly improves the quality of care across the U.S. and worldwide. Still, it is not uncommon for drugs to pass through the preliminary screenings for a drug with empirical animal studies to get to large-scale human trials and fail due to problems that could have been previously detected [10]. There is no doubt that there is room for improvement in drug screening, especially given that the process does not leverage some of the most exciting new innovations in toxicology: computational methods.

Currently, there exist computational toxicology models which are becoming increasingly useful in representing in vitro and in vivo studies [2]. However, these models remain excluded from the regulatory process for several reasons. One reason is simply that the predictive performance of the models is no better than the in vitro assays on which they are trained. Moreover, these models are centered around specialized endpoints, while live trials have more general outputs [3]. Perhaps the most critical reason which bars computational methods from the regulatory space, though, is their lack of interpretability. Computational methods, especially those that deal with machine learning, usually fail to justify their judgments through grounding in the underlying biochemical properties. This concept of interpretability could be subdivided into two categories, which we will refer to as "direct interpretability" and "high-level interpretability." Direct interpretability refers to contextualization within biological, chemical, and physiological principles, while high-level interpretability refers to explanations which are easily understood by users due to posited relationships based on high correlations for structural features.

In navigating these obstacles, it is important to recognize that performance can be improved only so much, especially given that computational methods rely on imperfectly collected data from live trials. Thus, while it is obviously critical to strive for strong performance compared to the field as it stands, it is not our current criterion of contribution within the field. Additionally, when considering that the level of complexity for an integrated human physiological model would be extraordinarily computationally expensive, among other issues of implementation, we do not believe that there are many gainful strides to be made with a more generalized system. We feel that the largest innovations awaiting the field of computational toxicology most likely are with regards to improving their interpretability. A proof-of-concept toward a computational model that can clearly explain the chemical mechanisms behind toxicity would strongly contribute to the field by demonstrating the growing potential of computational models to be used in drug regulation and screening.

With these components explained, we have enough information to give our research question in full: how can a computational model for toxicity prediction be developed with novel techniques to enhance interpretability, while retaining high accuracy? In answering this question, we will first move to survey the relevant literature on computational toxicology in general, with a special focus on interpretability metrics.

1.3 The Tox21 Challenge

Created by the FDA in 2014, the Tox21 Challenge brought together a global community of data scientists to apply machine learning techniques as a method for predicting drug toxicity. The Tox21 Challenge called on researchers to use structural data of chemical compounds to predict their interactions with biological systems, especially with regards to toxicity. These endpoints, presented below, are all in vitro and were the result of high-throughput assays. The dataset contains SMILES string representations of about 8000 molecules can be accessed through the DeepChem Python library.

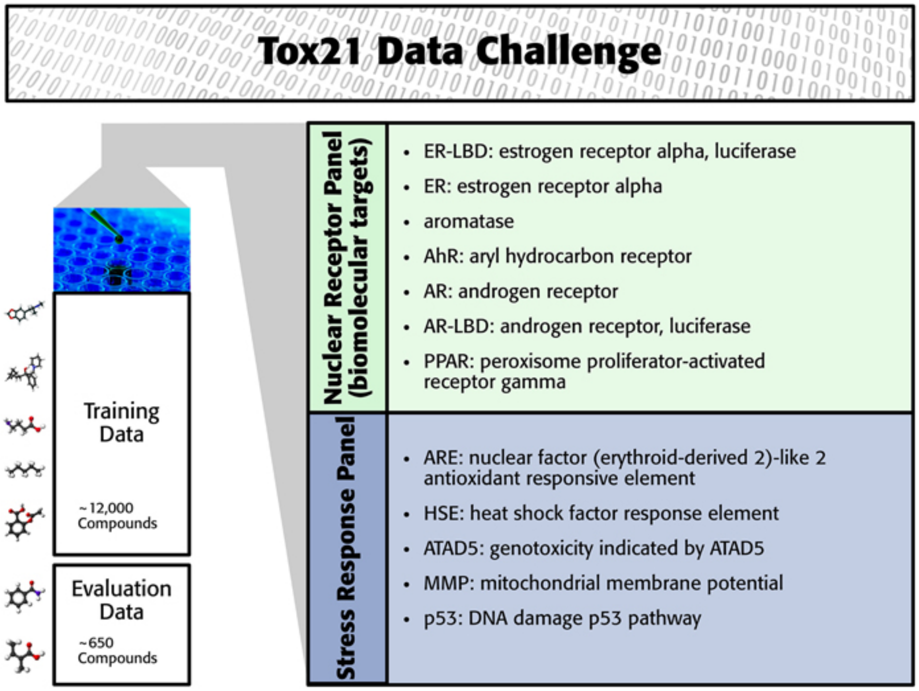
Tox21 Data Challenge
Training Data
~12,000 Compounds
Evaluation Data
~650 Compounds
Nuclear Receptor Panel (biomolecular targets)
• ER-LBD: estrogen receptor alpha, luciferase
• ER: estrogen receptor alpha
• aromatase
• AhR: aryl hydrocarbon receptor
• AR: androgen receptor
• AR-LBD: androgen receptor, luciferase
• PPAR: peroxisome proliferator-activated receptor gamma
Stress Response Panel
• ARE: nuclear factor (erythroid-derived 2)-like 2 antioxidant responsive element
• HSE: heat shock factor response element
• ATAD5: genotoxicity indicated by ATAD5
• MMP: mitochondrial membrane potential
• p53: DNA damage p53 pathway

Figure 1: Summary of the Tox21 dataset [11].

Due to the content of the Tox21 Challenge dataset, containing the toxicity classifications of thousands of molecules, it was the crucial data source on which the team based its investigations into toxicity prediction. Many other researchers have contributed models to the challenge, which were also of importance to the team in the area of model analysis. The below table provides detail regarding the class splits for the various assays included in the dataset. For each assay, class imbalances are clearly present such that the significant majority of molecules are not toxic for a given indication. Molecules are considered toxic for a given assay with an activity score deemed to be sufficiently high as to be conclusive, while considerations of no toxicity are usually reserved for molecules with activity scores of zero. Scores were determined based on the fit of the half-maximal activity concentration (AC50) to a logarithmic form. It should be noted that there are a variety of different molecular classes within the Tox21 dataset, ranging from small molecules to relatively large carbohydrate molecules. The domain of applicability for models developed includes the whole set, which notably preserves the inclusion of some salts.

Endpoint	Count	# Not Toxic	# Toxic	% Toxic
NR-AR	7265	7024	241	3.32%
NR-AR-LBD	6758	6566	192	2.84%
NR-AhR	6549	5937	612	9.34%
NR-Aromatase	5821	5583	238	4.09%
NR-ER	6193	5563	630	10.17%
NR-ER-LBD	6955	6688	267	3.84%
NR-PPAR-gamma	6450	6301	149	2.31%
SR-ARE	5832	5067	765	13.12%
SR-ATAD5	7072	6860	212	3.00%
SR-HSE	6467	6172	295	4.56%
SR-MMP	5810	5072	738	12.70%
SR-p53	6774	6420	354	5.23%

Table 1: Distribution of toxic and nontoxic molecules present for each toxicity endpoint in Tox21 Challenge.

1.4 DeepChem

DeepChem is a powerful open source framework for deep learning in the field of chemistry. Made publicly available in 2017, DeepChem makes use of Google TensorFlow to create neural networks for deep learning. It also makes use of the RDKit Python framework for operations such as creating molecular graphs out of SMILES string representations of molecules. Among the features it offers include featurizations of SMILES strings, including ECFP (Extended-Connectivity Fingerprints), Graph Convolutions, and Coulomb Matrices. These were the primary descriptor sets considered due to their easy availability in the DeepChem framework. Comparison models which were previously developed are often based in this framework and thus pri descriptor sets. In addition to the deep learning components of DeepChem, the framework also provides a large amount of chemical datasets, comprising over 500,000 chemical compounds. Much of the data that the team used was from the DeepChem-provided dataset, along with the Tox21 challenge dataset mentioned previously.

1.5 MoleculeNet

One of the many challenges associated with trying to improve molecular machine learning has been trying to determine whether new methods have improved upon the efficacy of the old ones. MoleculeNet was created with the intent of finally solving this problem. It establishes a standard benchmark that can be used to properly compare any improvement caused by new algorithms being tested. It does so by curating many public datasets and establishing evaluation metrics using them. It also offers open-source implementations of many published learning algorithms, built off of the DeepChem open source library. The features offered by the MoleculeNet project proved to be of great help to the team, as it was used to compare whether the team's proposed methods improved upon previously published and tested ones.

Part of MoleculeNet's work was the developement of baseline models of different architectures for example the deep learning methods described in sections 2.1 on 9, 2.2 on 9, and 2.3 on 10. They optimized hyperparameters such as numbers of layers and training times for multiple datasets including the Tox21 challenge. As of the time of this paper's writing, we have used the most updated hyperparameters that they provide.

2 Literature Review

2.1 Deep Neural Networks

Deep Neural Networks (DNNs) are a machine learning technique that are based on artificial neural networks with many layers consisting of a high number of neurons. DNNs have been one of the more popular methods in the machine learning community and have performed well on tasks such as speech recognition, computer vision, and other artificial intelligence applications. The goal of neural network learning is to create functions that map an input vector to an output vector and adjust the network weights so that the input-output mapping has a high predictive power on future data. The mapping is parameterized by weights that are optimized during a learning process. As opposed to shallow networks which only contain one hidden learning layer and a few hidden neurons per layer, DNNs contain many hidden learning layers with many neurons each. This layout contrasts with traditional artificial neural networks, which only use a few neurons. Deep neural networks are an improvement upon classical neural networks in that the number of input descriptors that could be handled is greater. This change allows models to retain valuable predictive information, thereby assisting the DNN in capturing all possible aspects of the input vector [11]. Notably, a DNN approach proved to be the best-performing model in the original Tox21 competition [11].

2.2 Graph Neural Networks

Graph Neural Networks are forms of machine learning techniques which can operate on graphs. While there are many variants of this class of models, they all follow a similar structure. They all have an encoder, which generates node embeddings for a local neighborhood of nodes, a decoder, which translates these embedding back into the graph format, and a similarity metric, which is used to compare nodes in a graph [12].

The encoding step takes a node in a graph and compresses information about it and its neighbors into a lower dimensional vector which ideally preserves the properties of the original graph, and the decoder undoes this step. The size of this neighborhood and the function used to aggregate nodes at each step is user defined. Optimizing the encoder function is achieved through training using the similarity metric, since nodes which are similar in the graph must also have similar embeddings. Once the encoder and decoder are trained from a training set of graph representations, they can be applied to find either new relationships, or edges, between different graphs, or to classify a node in a different graph, depending on how the encoder is optimized.

These models have the advantage of being able to operate on multiple domains of information. Graphs are an extremely versatile data structure, as nodes and edges can represent various elements and their interactions. One group was able to leverage this to predict polypharmacy side-effects using information about all the various drug-drug, drug-protein, and protein-protein interactions, with a high degree of accuracy [13]. To achieve this level of accuracy, however, such information has to be available, therefore these models are not so useful where data for new inputs is sparse. However, this is a problem faced by any model using multiple domains of data. These models are also potentially very interpretable, thanks to their decoder. If the task is predicting edges, which represent interactions that are being predicted by the modeler, then the physical interpretation of a prediction will be apparent based on the kind of interaction predicted by the decoder.

2.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are machine learning techniques that work by convolving (or using a preset function to layer, shrink, and modify) their input and then sending it through a variety of different layers. There are many different ways to arrange these layers, but the fundamentals remain the same from model to model. Each model will start with a convolution layer, which will take a convolution kernel which assigns every pixel a pre-set weight, traverse the input image pixel by pixel, and use the kernel to create a new image based off of its weights. This first convolution layer primarily detects the low-level features, namely curves and edges [14].

CNNs also use a pooling layer, whose purpose is to reduce the resolution of the feature maps generated by the convolution layers. By reducing the resolution of the output maps, the layer's kernels are able to detect more abstract features and improve the model's overall sense of the image. The third layer used is known as the fully-connected layer. Much like the layer's name would suggest, the fully-connected layer takes all of the neurons generated by the previous layer, regardless of what type of layer it might be, and connects them to all neurons of the current layer. In doing so, the model creates global semantic information which helps it produce a more accurate output image.

The backpropagation algorithm is the standard algorithm for training CNNs, which analyzes errors and loss previously generated and calculates new weights for the different parameters accordingly. The biggest strength of a CNN is its incredible ability to classify and label various images, which is a capability that we could use in identifying various compounds and chemical descriptors. Another strength is that the various layers of the CNN complement each other, which improves total accuracy of the model. With these strengths come weaknesses, one of which is a CNN's tendency to overfit. When a model overfits, it essentially "memorizes" its training dataset and produces inaccurate and poor results when the input data is different [15].

Recent research has worked towards generalizing CNNs to operate on graphs. The classic CNN operates on Euclidean data with grid structures; most commonly, images [16]. Grids can be considered a subset of graphs that are lattice-shaped and more rigidly defined. Generalizing CNNs to operate on all graphs, which can be non-Euclidean and more flexible in application for data representation, is greatly applicable to the development of models based on molecule representations, since a graph structure based on bonds between atoms is an intuitive representation of a molecule.

2.4 Interpretability Methods

Motivation

Machine learning has dominated classification efforts in recent years, but wielding the technology is a double-edged sword for regulatory purposes. In terms of predictive power, machine learning is practically without peer, allowing for unprecedented accuracy in classification of toxicological effects. However, this predictive power comes at the expense of interpretability. Interpretability is the ability of the model to explain how it identifies output classifications to a human in an understandable way. Many of the strongest machine learning methods are inherently "black box" mechanisms which produce output from input in a way that cannot be fully understood. The black-box problem, a fundamental one in machine learning not unique to toxicology, is of critical importance in the field to clearly identify the relationships that govern chemical activity. Without the ability to interpret these strong models, they cannot be completely trusted by regulatory bodies.

In 2007, the Organization for Economic Co-operation and Development released a since-revised comprehensive document governing the validation of predictive models [17]. This document identifies five main principles for the regulatory acceptance of predictive models. Accordingly, they should have: a defined endpoint, an unambiguous algorithm, a defined domain of applicability, appropriate measures of goodness of fit, robustness, and predictive power, and, if possible, a mechanistic interpretation. While machine learning methods can satisfy some of these requirements, there is a significant amount of work to be done to improve interpretability. Effective methods for interpreting black-box predictions will significantly advance drug development and regulation via an increase in model trust, which will improve overall public health as well. For this reason that we wish to center our research around novel techniques for the interpretability of machine learning models.

The first solution towards interpretability is the simplest: use the simplest model architectures. A subset of algorithms including logistic regression and decision tree all satisfy some properties that make them inherently easy to understand. In a logistic regression model, there exists a monotone relationship between each feature and the final prediction. That is, either the final output only increases as a feature value increases, or the final output only decreases as a feature value increases. In a decision tree model, the architecture explicitly singles out those features that account for the greatest variance in the final output. Given the choice between models of similar performance, the simpler architecture will yield the greater interpretability.

The difficulty in tackling the issue, however, lies in balancing the tension between predictive power and interpretability. Models of increasing predictive power used deep learning approaches of increasing complexity. A human's ability to follow a classifier's reasoning required some degree of simplicity. The solution lied in model-agnostic interpretability methods, methods that produced explanations of model behavior post prediction without any requirements on the classifier's underlying architecture. Additionally, in some contexts, descriptors which appear often in various models are examined with regard to biological interpretation regardless of architecture. This expert rationale may form an import component of interpretability.

Local Interpretability Methods

One solution is local interpretability methods, which tend to be model-agnostic, meaning they require no specific model architecture. Rather than deriving an interpretable function to explain the entire model, such as the coefficients in a logistic regression model that are relevant to the entire dataset, an alternative approach to interpretability is to explain individual inputs to the model instead. A highly nonlinear, complex decision boundary is more likely to appear smoooth and linear when localized to a certain region.

This is the intuition behind LIME, a major breakthrough in local interpretability methods [18]. For a given input x to a model f, this algorithm attempts to return an explanation, an interpretable function g, which satisfies two conditions: it is simple, and it is locally faithful. These two conditions are required to obtain a good explanation for an input. Locally faithful means that the behavior of g must correspond to the behavior of f in x's vicinity. Clearly, for g to provide any insight at all into why the model outputs $f(x)$, it must be a function which behaves like f, at least in the vicinity of x. Complexity is a condition because not all interpretable functions are made equally. A linear function with thousands of variables is not as comprehensible as one with five, and the second would be preferred if it does not sacrifice too much local fidelity. The optimal function g, which LIME derives, is therefore the function which finds a balance between maximizing faithfulness and minimizing the complexity of the function.

This algorithm provides its users with a direct insight into the internal workings of the model, which bolsters confidence in predictions and opens the door to mechanistic interpretations, which are a must for regulatory compliance. By examining the relationship of the weights in the function returned by LIME, a user can derive a mechanistic interpretation by translating the weights to the real chemical features they correspond to. While determining the validity of the relationship expressed by this function does require expert knowledge, this is far better than accepting the output on blind faith.

Offshoots of this work continued developing the concept of local interpretability. Instead of forming a local surrogate model, Lundberg and Lee used game theory to treat features as players and determines their contributions towards the final prediction [19]. In "Anchors: High-Precision Model-Agnostic Explanations", the original authors of LIME propose Anchors, an algorithm that uses reinforcement learning and graph search learning to identify decisions rules unaffected by changes in other features [20]. The authors demonstrated its ability to aid users in predicting its classifications of unseen data in comparison with linear-based interpretability methods. It shares a reliance with the original LIME paper on observing changes in model behavior based on perturbations to data.

The papers listed above yield high fidelity explanations for model behavior on instances of data, and their strength lies in their model agnosticism. This includes models ranging from decision trees to convolutional neural networks. However, implementations of such work are scare for models that handle some of the complex data representations associated with chemical data. That is, little work exists for the interpretation of models handling molecular graphs.

Graph Convolutional Neural Network Interpretablity Methods

Pope and colleagues produced one major effort that generalizes existing interpretability methods for convolutional neural networks to graph convolutional neural networks [21]. After all, image data handled by regular convolutional networks are really instances of lattice-shaped graphs. Their methods create heat maps over the nodes of graphs using five methods including gradient-based methods, class activation mappings, and excitation backpropogation. Finally the paper evaluates its explanations by fidelity, contrastivity, and sparsity, three interpretability metrics defined respectively as the loss in accuracy suffered by the removal of features, the distance between explanations of opposite classes, and the ratio between the number of nodes in a graph highlighted for all output class to the number of nodes in the entire graph.

By incorporating these techniques, described in the context of general ML as well as to some extent in toxicology literature, we extend the work of methods previously applied to the Tox21 initiative for increased impact. Namely, we focus on LIME for models based on molecular fingerprints [18] and on Pope et al.'s work for models based on molecular graphs [21].

2.5 Model Accuracy Parameters

The most important metric to measure when developing machine learning models is accuracy. If a model is particularly inaccurate, then there is no utility associated with it, as there is no confidence in the results that it outputs. The primary measure of accuracy is the number of correct identifications divided by the total number of data points in the test set. This is acceptable for most machine learning models, but in some cases, it is possible to have high accuracy by this metric and still have a bad model. This can occur if the representation of a positive and negative model is vastly unequal. Unequal distribution of the classes in the training set can lead to the model training in such a way that maximizes accuracy by treating every input as either positive or negative regardless of the input's characteristics. If a model is trained on data where 90% of the data points are in the positive class, then the model can achieve 90% accuracy by classifying everything as positive. The problem is that the negative class is identified incorrectly 100% of the time.

Additional accuracy metrics that can be used are the ROC curve (receiver operating characteristic curve) and the AUC (Area under the ROC curve). The ROC curve is a curve that shows the true positive rate (x-axis) versus the false positive rate (y-axis) for the model being tested. In the above example, the false positive rate is 100%, and the true positive rate is 90%, meaning that the point on the ROC curve is (1,1). The model is manipulated to perform with differing true positive rates, and the false positive rate is measured, and the curve is generated. Once the curve is generated, then the AUC is found. An AUC of 1 is defined as being a perfect model, and an AUC of 0 shows a model that classifies everything incorrectly. The goal of the model is to approach an AUC of 1; as it gets closer, the number of false positive rate will be shown to be lower for various class distributions. Balanced accuracy, the average of sensitivity and specificity, can serve as another option for addressing model performance.

2.6 Molecular Representation

Though molecules are commonly drawn as two-dimensional structures, collections of letters and lines, they are in fact three-dimensional assemblies of atoms which are always in motion. The vibrations of molecules may be approximately calculated using computational chemistry techniques. However, in considering how to "draw" molecules for the use of a computer program, it is not always useful or desirable to fully describe a molecule in the context of its wavefunctions or quantum effects.

For many cheminformatics applications, small molecules are represented via SMILES (simplified molecular-input line-entry system) strings. Invented in the 1980s by Dave Weininger, the flexible and compact formatting of SMILES strings have made them ubiquitous in chemical databases (¿2000 citations on the original paper as of the time of writing). Notably, SMILES relies on a graph-theory approach to segmenting molecules into their component branches before assigning sets of symbols to each branch. Despite their merits and common usage, SMILES strings may be unwieldy due to their lack of mandatory stereochemical information as well as imprecision in SMILES syntax [22]. A variety of different strings may be used to represent the same molecule, particularly for ring-containing systems. Additionally, it is difficult to ensure that SMILES strings refer to valid molecules at all without checking them. This is to say that the space of all possible SMILES strings is composed of many non-molecules [23].

As such, there have been many iterations of improvements on the SMILES formatting to correct against these issues. Efforts have been made by various groups of chemists to canonicalize SMILES strings, such as those of Daylight Chemical Information Systems (founded by Weininger himself) [24] and of OpenEye. The Daylight algorithm is commonly implemented since it was written by Weininger, but does not incorporate a way to handle stereochemistry. OpenEye's algorithm is of note since it is used by many governmental agencies, such as NIH, for the handling of their chemical databases. However, none of these canonical algorithms has been established as a standard, and it is generally necessary to convert SMILES strings of unknown origin to a canonical form using a given software package, even though it may not match up with other databases effectively. There have additionally been attempts to universalize chemical line-entry data apart from SMILES. The InChI ("International Chemistry Identifier") format was developed by the International Union of Pure and Applied Chemists (IUPAC) in the mid-2000s as a standard for computational nomenclature. However, they are not always easy to use for computational purposes, and as such are generally limited to molecular identifiers. Unlike SMILES, there does exist a standard, open source algorithm for the generation of InChI representations [25]. Fortunately, there have been relatively successful efforts described to interchange between InChI and SMILES [25].

As explained at the outset of this section, real molecules are not static objects. However, they do lend themselves rather nicely to being represented in graph formats. After all, atoms may be thought of as the nodes in an undirected graph, with various types of edges corresponding to various types of bonds [26]. It is easy to conceptualize this in two dimensions, and the extension to three dimensions is not much more difficult. From here, though, it is usually necessary to transform such graphs into arrays for computer readability [26]. (While there are salient questions of image recognition for molecular visualizations, those will be considered at a later point.) This transformation requires algorithms which can map atoms to a given order, with edges defined in between them based on the bond order of the molecule itself [26]. A corollary to this, which often complicates the processing of molecules, is that these arrays are order-dependent even though the originating graphs are not [26]. While these are useful properties, graphs generally fail to capture more complex bonding types (i.e. three-center bonds) and may be memory-inefficient [26].

There are also methods by which to encode chemical information based on non-atomic information, known as molecular descriptors [27]. Otherwise known as molecular fingerprints, these representations encode structures based on the presence or absence of specific functional groups. There are, though, many different types of fingerprints which account for molecular structures in a variety of ways. A popular format, the Extended Connectivity Fingerprint (ECFP), are circular in nature and are quick to calculate by using the Morgan algorithm as a standard for operating [28]. The primary application for such fingerprints is in structure-activity relationships (SARs) which use a group of structurally related molecules [28], enabling ECFPs to exhibit greater utility than they might across a grouping of unrelated molecules [29]. Fingerprints may also cover chemotypes (highly structurally similar classes of drugs) as a simplifying technique to cover the chemical space which arises most often in practical usage. Using molecular fingerprints to track similarity, Drwal and colleagues described the implementation of a naive Bayes classifier approach successfully to learn on the Tox21 dataset [30].

3 MoleculeNet Metric Comparison

3.1 Background

Area Under the Curve (AUC) is a metric that is commonly used to benchmark prediction models where the curve is typically a Receiver Operating Characteristic (ROC) curve [31]. The ROC curve plots the true positive rate against the false positive rate at a variety of classification thresholds, so a classifier with an AUC-ROC of 1 would be ideal. AUC-ROC is the metric MoleculeNet [32] used to compare different classifiers for Tox21 as suggested by [33]. AUC-ROC is agnostic to class imbalances [34], which is a property of Tox21. In Tox21 there are many more non-toxic (negative class) compounds than there are toxic (positive class) compounds. As a result of an unbalanced data distribution, classifiers built on Tox21 can be less sensitive to the toxic class which can cause a higher rate of false negatives in the classification [34]. In the practice of toxicology, false negatives are especially dangerous as it is the case where a compound is in fact toxic but is not identified as such. Since AUC-ROC is not dependant on a balanced dataset, it is a justified metric to compare prediction models; however, the metric is negligent to the importance of sensitivity in toxicity prediction.

A component of improving a model's interpretability is studying how to report the model's performance completely. Performance metrics we considered were area under the curve (AUC), accuracy, sensitivity, and specificity. We collected performance metrics for the models we explained and observed that each metric gave a different impression of performance compared to the AUC-ROC. For further investigation, we collected the four metrics on every Tox21 model in MoleculeNet [32] and again observed a significant variation for each metric between models. This experiment reveals that AUC-ROC does not predict other metrics for Tox21, and comparisons of model performances would benefit with the inclusion of other measures.

Accuracy is the measure of the model's ability to correctly identify toxic compounds as toxic and non-toxic compounds as non-toxic.

$$Accuracy = \frac{true\,positives + true\,negatives}{total\,predictions} \tag{1}$$

Class specific accuracy separates the classes in order to calculate the accuracy. Sensitivity is the positive class accuracy, how well the model can identify a

AUC-ROC vs Sens.	AUC-ROC vs Spec.	AUC-ROC vs Acc.
0.3304	0.2374	0.1625

Table 2: P-values for AUC-ROC againt sensitivity, AUC-ROC against specificity, and AUC-ROC vs Accuracy for the Tox21 Benchmark models.

compound as toxic. Specificity, on the other hand, is the negative class accuracy, how well the model can identify compounds as non toxic.

$$Sensitivity = \frac{true\,positives}{true\,positives + false\,negative} \tag{2}$$

$$Specificity = \frac{true\,negatives}{true\,negatives + false\,positives} \tag{3}$$

3.2 Metric Comparison Method

As MoleculeNet [32] utilized the open source package DeepChem [35], we used this software to recreate the Tox21 models benchmarked in MoleculeNet [32]. Although the architectures were provided in DeepChem [35], the code in MoleculeNet [32] is not available, and so the AUC-ROC values displayed in Figure 1 are slightly lower than those in MoleculeNet [32]. We then applied the evaluation method to the models we operated on, graph convolutional network and multitask classifier, to collect AUC-ROC, accuracy, sensitivity and specificity.

3.3 Discussion

In this section, we report our observations from the metric comparison experiment.

Figure 2 illustrates how models with comparable AUC-ROC can have relatively large, inconsistent differences in other metrics. The models are organized left to right in ascending order by AUC-ROC, yet the other metrics do not follow this trend. For example, models with higher AUC-ROC do not necessarily have higher sensitivity. Another observation is the relative score of metrics differ between models arbitrarily, for example, accuracy is not always greater than AUC-ROC and the size of this discrepency is not the same between the models. Table 1 quantifies the lack of correlation between AUC-ROC and other metrics with P-values that are not statistically significant for $p < 0.5$.

From these results we realize there is an unpredictable difference within a model's accuracy and class specific accuracy relative to AUC-ROC, additionally the magnitude and direction of the difference between metrics is not implied by AUC-ROC. It is important to know the other metrics in isolation because they explicitly describe different properties of a model's performance.

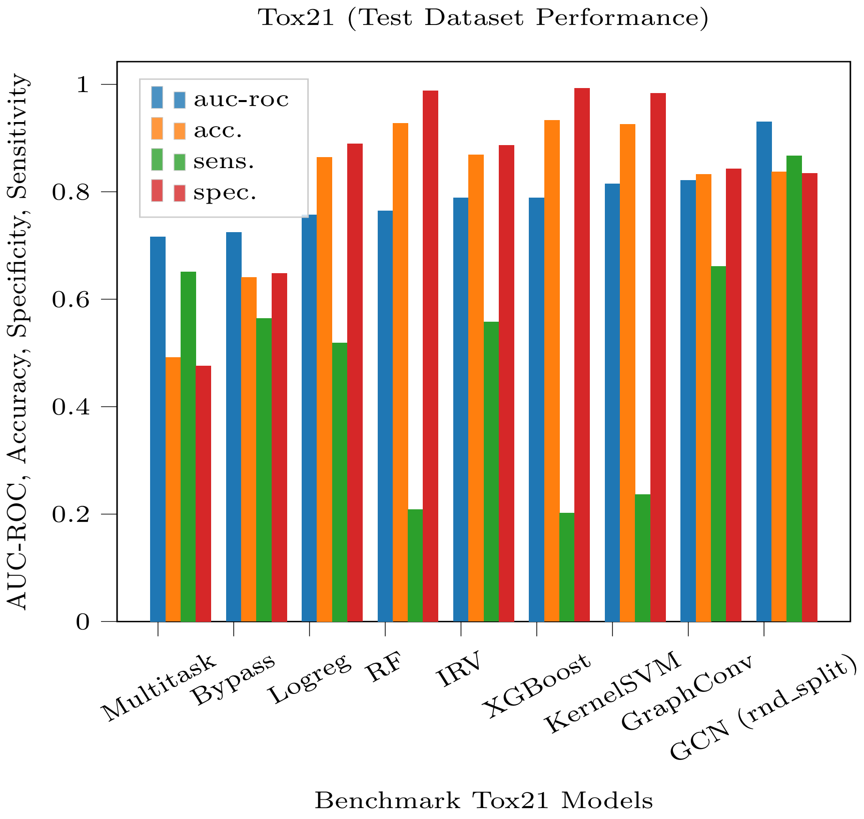

Figure 2: Comparison of performance metrics for Tox21 Models included in MoleculeNet.

AUC-ROC is a meaningful metric to measure the performance of a Tox21 prediction model, but it is not complete by itself. Due to the substantial and unpredictable variation of a model's metrics we recommend including accuracy, sensitivity, and specificity to supplement AUC-ROC when reporting metrics.

4 Evaluation of Model Explainers

4.1 Background

Expanding on [21] described in the literature review, evaluations were conducted on the fidelity, contrastivity, and sparsity of the explanations. Recall that fidelity is the difference between the model's performance before occluding influential features as determined by the explainer, and after occluding these features. The features with weights greater than a certain threshold are selected to be removed. A strong explainer is expected to have a high fidelity score, as this means that the explainer is correctly identifying the key features that the model recognizes as important to its classification decisions. Contrastivity is measuring the uniqueness of features identified for the different classes the model can assign. A high contrastivity is a property of a strong explainer because it demonstrates the ability of the explainer to create a clear set of features responsible for the specific classifications of the model. Sparsity is a ratio of the number of relevant features as determined by the explainer to the total number of features, a useful metric to monitor.

4.2 Explainer Methodology

To create these metrics, the methodology used was adapted from the approach mentioned in [21].

One major adjustment to this methodology was the modification to fidelity as a difference in accuracies before and after the model has features occluded. The motivation for this was due to preliminary results where occlusion of features noted as relevant by the explainer either maintained the same accuracy or even improved the accuracy of the model. This seemingly contradicted what would be expected from feature occlusion, but was suspected to be a result of an unbalanced dataset. As shown in section 1.3 on page 6, the dataset is dominated by non-toxic class samples, so the model may be prone to predicting a non-toxic over a toxic classification, which would result in this unusual accuracy behavior. Observing this, we adjust the definition of fidelity to be the difference of AUC-ROC before and after the occlusion of features. We extended the study of [21] by applying the evaluation metrics to all 12 tasks of Tox21.

We also made adjustments when extending these evaluation metrics to the LIME explainer, operating on bits in a feature array as opposed to nodes.

Method	Contrastivity	Sparsity	Fidelity
CAM-GradCAM	1±0.000151	0.929±0.0124	0.15±0.079
EB	0.444±0.0486	0.675±0.0561	0.21±0.0604
GradCAM-avg	0.487±0.0475	1±4.32e-05	0.208±0.066
Gradient	0.0139±0.0031	0.992±0.00249	0.438±0.0647
cEB	0.912±0.021	0.751±0.0422	0.172±0.0997
LIME	0.108±.0558	0.0780±0.0461	0.0

Table 3: Evaluation metrics for Tox21 for all graph explainers and LIME. Fidelity was calculated using a threshold of 0.05.

We realized that LIME produced a different distribution of values to quantify the importance of certain features and performed experiments which impacts the calculation of fidelity. To address this we included a LIME fidelity calculated at a lower threshold.

4.3 Discussion

Figure 3 on page 21 displays the results of the preliminary experiments, the mean and standard deviation of fidelity across 12 tasks calculated using accuracy. The substantial variation in fidelity, even taking on negative values, prompted the investigation into using AUC-ROC to calculate fidelity.

Figure 4 on page 22 presents the mean and standard deviation of fidelity calculated using AUC-ROC. Comparing Figure 4 to Figure 3 suggests that AUC-ROC is a more stable way of calculating fidelity for Tox21.

The contents of table 3 on page 20 display the mean explainers evaluation metrics for the 12 tasks for each graph method and LIME. The fidelity for LIME is 0.0 because at the threshold used to occlude features was too great for the values LIME set, and so no features were excluded. This is because the applicability domain was kept open to all molecules within the Tox21 dataset.

5 Electrostatic Potential Model

5.1 Background

An important facet of our research is the development of a model that takes in a novel input format. Normally, it is assumed that the structure of a molecule is important in predicting its toxicity. Certain arrangements of atoms and features are often noted as contributing to toxicity. In our model, we decided to include additional information for molecule representations in the form of partial charge information, which form the basis for electrostatic potential (ESP) maps.

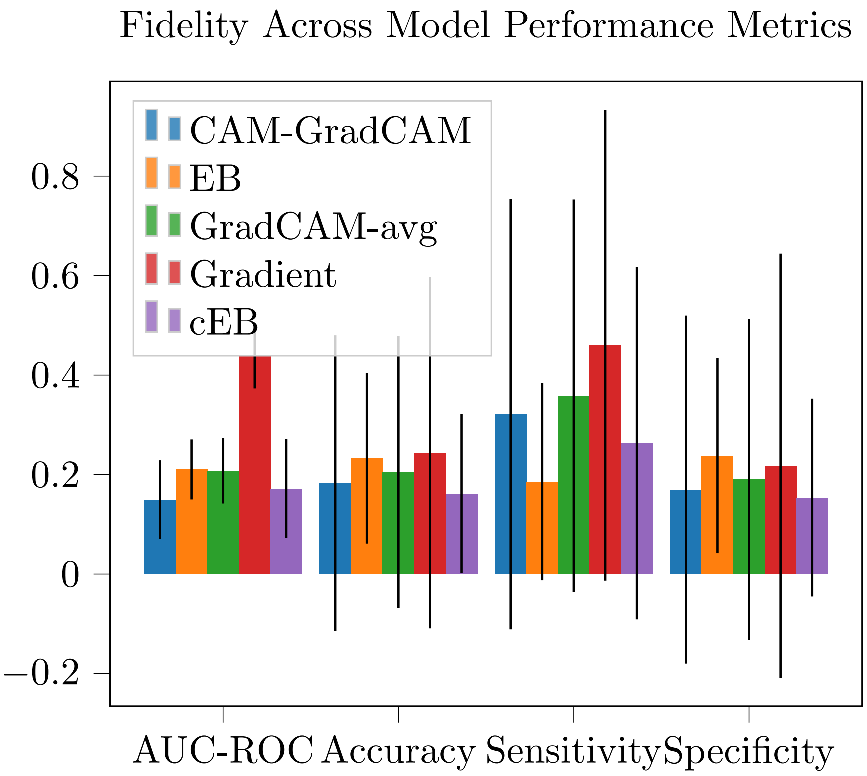

Figure 3: Fidelity, or loss in model performance due to removal of features highlighted by interpretability methods, averaged over Tox21 dataset. Error bars are standard deviations. Colors correspond to interpretability methods.

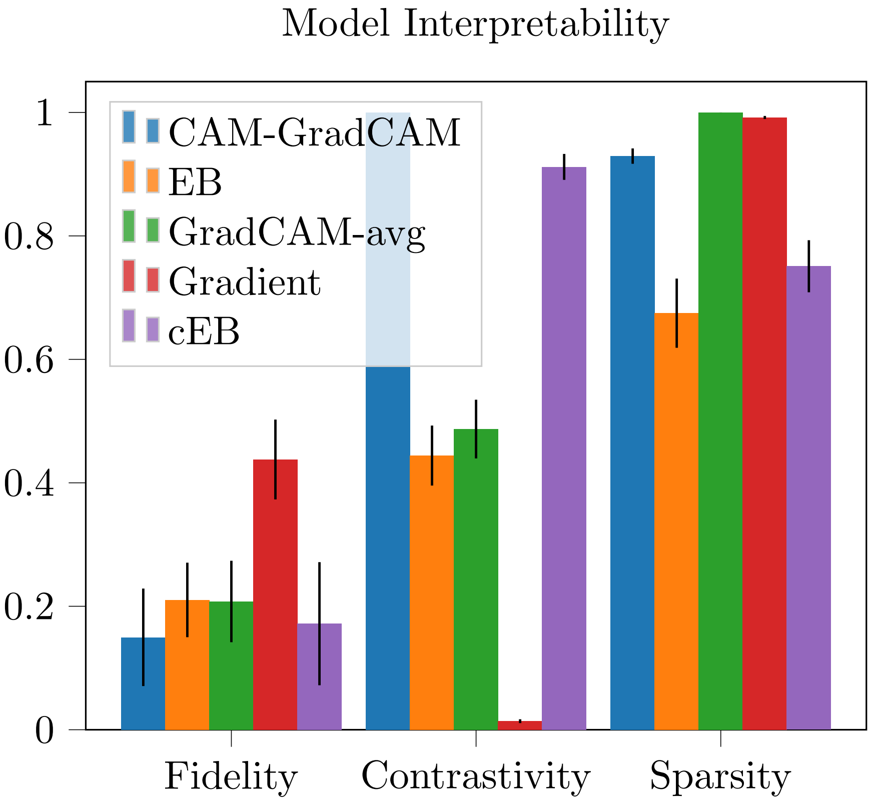

Figure 4: Model interpretability metrics averaged over Tox21 dataset. Fidelity is redefined as loss in AUC-ROC rather than loss in accuracy. Error bars are standard deviations. Colors correspond to interpretability methods.

We expected that accurate charge-based data would serve as an improvement to current models based on the commonplace use of simple two- or three-dimensional structures fed into models. This was desired as opposed to Gasteiger charges or other straightforward partial charge models in part due to accuracy, as well as the potential novelty of a simplified workflow that would ideally approximate the improved resolution of DFT-based charge prediction while avoiding its computational expense. The way molecules influence biological systems in part depends on molecule dynamics, and could be useful in toxicity predictions. Thus, developing a representation for molecules' ESP fields to be used as an input for our own model is a novel direction in computational toxicology.

5.2 Dataset Construction

The initial data used by models in the Tox21 challenge were recovered in SMILES form from the deepchem library [35]. These were transformed into a multi-molecule .sdf file using the CACTUS service provided by the National Institutes of Health. There were minor errors at this step (9 molecules out of 7831) for molecules containing macrocycles and/or bridged cycles. Structures for these 9 molecules were manually recovered using PubChem's structural similarity search function. Using the OpenBabel package, this file was split into individual molecular files (.mol file extension as an output from this step).

In order to obtain inputs in the correct form, as PQR files, the SMILES strings from the Tox21 database went through a multi-step conversion process. First, they were converted into canonical SMILES form, specified by InChI. These could be used to retrieve chemical IDs and SDF files from the PubChem database. OpenBabel [36] was used to convert the SDF files into MOL files. From here, the ESP-DNN package [37] was used to generate partial charge data for the molecules (n=7831) in the dataset. ESP-DNN is notable since it is a computationally inexpensive method which has been shown to yield electrostatic potentials at a comparable level to density functional theory (DFT) [37]. While we initially considered the use of DFT methods for the generation of quantum chemical data [38], we decided against this largely due to constraints on computational resources available. As such, we sought to use it for our purposes as an efficient method for predicting partial charges on given atoms. At the time of writing, it appears that this method has not previously been used for the development of any computational toxicology model.

ESP-DNN was used to convert the MOL files into the PQR files that can then be used to predict point charge densities. In total, 5308 molecules were in the dataset at this step, due to errors in conversion processes and queries removing several molecules. Using OpenBabel, these PQRs were reconverted to PDB files. Importantly, the reconversion was performed using the AssignBondOrdersFromTemplate command in OpenBabel with the original MOL files as templates, which was necessary since the PQR files themselves did not retain appropriate bonding information, particularly for aromatic groups.

5.3 Hypothesis Testing

In order to test our hypothesis that the addition of ESP fields would improve model interpretability without comprising model performance, we performed a comparison of two otherwise identical models, one model with the ESP fields and one model without. Due to the computational cost of the interpretability methods for molecular fingerprints combined with the lesser predictive power and interpretability of models that use them, we restricted our analysis to the molecular graph data representation. We added ESP fields to the molecular graph representations by appending the partial charges for each atom in a molecule to the molecule's feature matrix.

Working with molecular graphs with and without ESP fields, we selected the same model architecture and hyperparameters from [21] for their favorable balance of high model performance with high model interpretability shown in section 4 on page 19. Based on our recommendations from section **??** on **??**, we measured model performance using AUC-ROC, accuracy, sensitivity, and speficity. We use the same methodology for model interpretability from 4 on page 19, using the interpretrabity methods and intepretability metrics from [21], but redefining fidelity as a loss in AUC-ROC rather than a loss in accuracy. For both models we used the same subset of molecules from the Tox21 dataset for which we could recover ESP fields in section 5.2, which we refer to as the Tox21 ESP dataset.

After collecting model performance and intepretability metrics for the two models and aggregating them over the Tox21 ESP dataset, we performed two-sample t-tests to determine the statistical significance of their differences.

	Accuracy	AUC-ROC	Sensitivity	Specificity
Dataset				
Training	8.3e-31	5.02e-56	2.05e-20	1.27e-27
Validation	8.1e-30	3.44e-39	1.45e-16	1.64e-26
Test	2.83e-30	8.75e-47	7.21e-19	1.62e-27

Table 4: P-values of two-sample t-tests comparing model performance of graph convolutional models with and without ESP data averaged over Tox21 ESP dataset

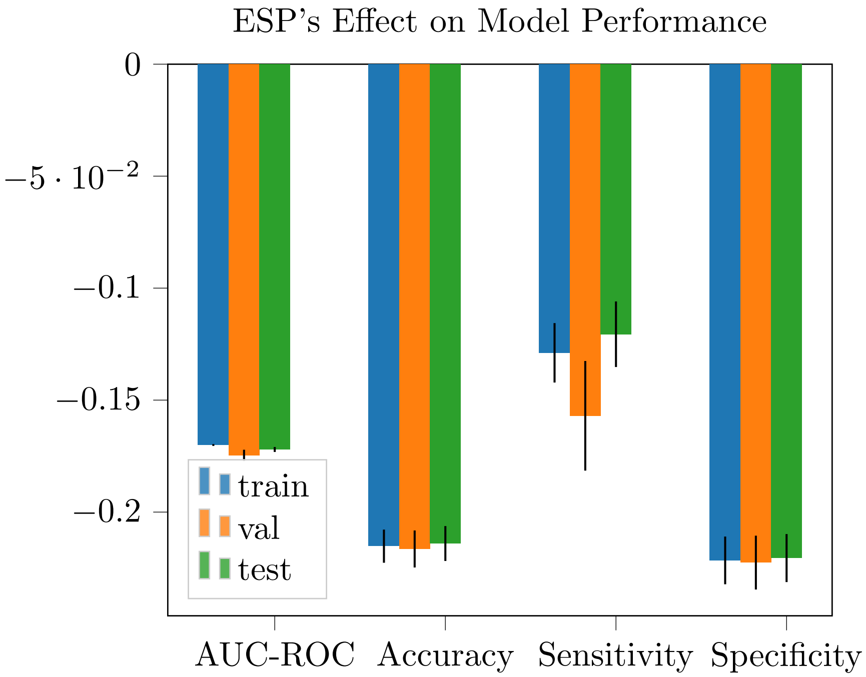

Figure 5: Change in model performance metrics of graph convolutional models due to addition of ESP data averaged over the Tox21 ESP dataset. Error bars are standard deviations. Colors correspond to performance on partitions in the data.

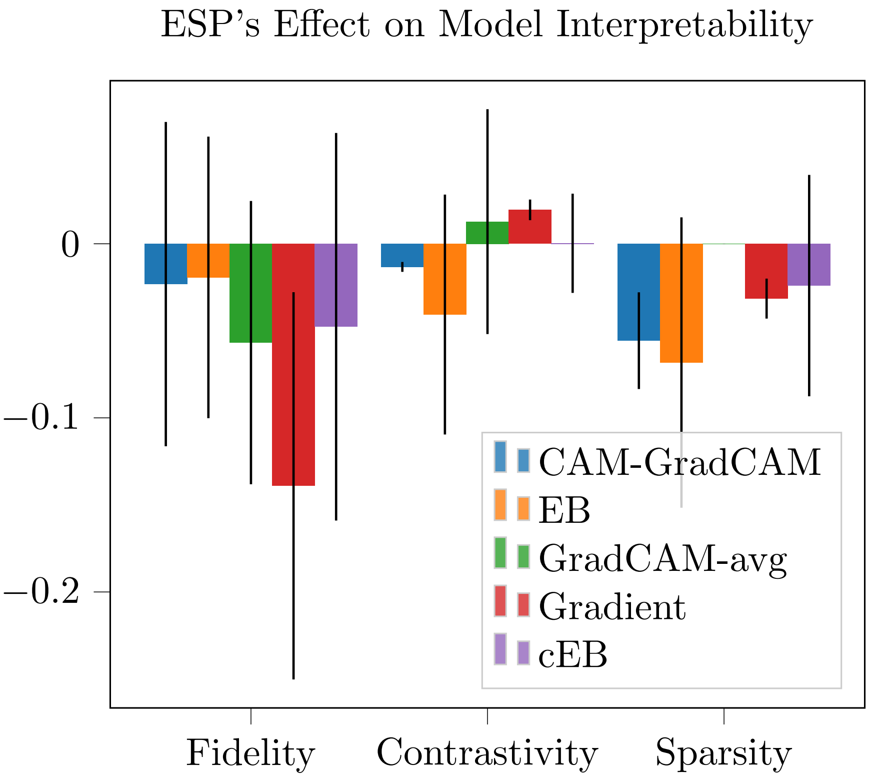

Figure 6: Change in interpretability metrics of graph convolutional models due to addition of ESP data averaged over Tox21 ESP dataset. Error bars are standard deviations. Colors correspond to interpretability methods.

Method	Contrastivity	Sparsity	Fidelity
CAM-GradCAM	9.84e-14	5.74e-07	0.398
EB	0.0532	0.00967	0.417
GradCAM-avg	0.502	0.39	0.0244
Gradient	1.08e-10	3.03e-09	0.000269
cEB	0.972	0.204	0.152

Table 5: P-values of two-sample t-tests comparing interpretability of graph convolutional models with and without ESP data averaged over Tox21 ESP dataset

5.4 Discussion

Figure 5 on page 25 shows that the addition of ESP fields reduced predictive power, and table 4 on page 24 shows that the differences were of statistical significance. Figure 6 on page 26 in combination with table 5 shows that the addition of ESP fields produced negligible change in interpretability metrics, with few significant results and none indicating a significant improvement for any of the graph-based interpretability methods.

Figure 5 on page 25 shows the difference in model performance between the model with ESP fields and the model without. The addition of ESP fields reduced model performance in training, validation, and testing on all metrics, and table 4 on page 24 shows that the losses were of statistical significance. Thus, the new data did compromise model performance of the graph convolutional neural network. However, its performance remained comparable to that of lower performing models mentioned in our evaluation of model performances in section 3.

Pressing onto interpretability, figure 6 on page 26 shows the difference in model interpretability between the model with ESP fields and the model without. The addition of ESP fields produced no significant improvement in fidelity, contrastivity, and sparsity for the explanations produced by each of the graph intepretability methods. Table 5 on page 27 shows that the few changes in interpretability metrics that were positive could likely be due to chance.

We believe that this impaired performance was in part due to errors in the estimation of the partial charges in our methodology. Based on a review of the input structures from the Tox21 ESP dataset, the geometries of the input structures appeared to be somewhat distorted (i.e. slight errors in bond angles). While intuitively it seems likely that the model would account for small deviations such as these, it may be the case that they factored more heavily than expected into the processing of the data. Future investigations should further consider the implementation of this or related ML-based charge prediction methods for the basis of generating augmented datasets.

6 Conclusion

Computational toxicology is an important field which will enjoy future growth, both in industry and academia, given the recognized need for better screening process in tandem with improvements in data analysis techniques. The drug trial process is often times long, expensive, and potentially dangerous for those that participate as subjects. Computational toxicology seeks to reduce the necessity of live drug trials by ruling out certain drugs based on their similarity to known toxic molecules. Our model will be built using past drug trials and results as training data, so that we can predict whether unexamined molecules will be toxic or not. Since the quality of our model is built on these previously conducted in vivo and in vitro trials, it is crucial that we are able to select and use high-quality data.

The ultimate goal of this project has been to create a model which can both make accurate predictions regarding toxicity within the human body and clearly explains the predictions it makes. Current computational models represent the toxicity of untested drugs fairly accurately, but we attempt to maintain these levels of accuracy in our model while avoiding the black-box paradigms of these existing models by incorporating features which provide context and interpretability for its decisions.

It was found that in addition to the overall model accuracy, the AUC-ROC was found to be a meaningful metric to be used within the realm of computational toxicology. The AUC-ROC is important as it identifies the model's tendency to have false positives in its predictions. However, AUC-ROC is not the only useful metric. Overall accuracy, sensitivity, and specificity are all also recommended for model assessment.

From the models that were developed, we found that the interpretability results provided by LIME for a bitstring model were significantly worse than the results given by the graph convolution network explainer. This is to some extent to be expected, as the input representation that LIME was attempting to interpret was imperfect, and could not be traced back to specific instances of present atoms in molecules, whereas the graph convolution network was able to identify each of the atoms in the molecule and rank their respective importance to the toxicity prediction. This resulted in the graph convolution explainer having significantly higher performance in fidelity and other interpretability metrics.

The other method that attempted to improve the interpretability of the models was to provide electrostatic potential information as the input for a model. The electrostatic potentials of each molecule in the dataset were calculated through a pipeline starting with the SMILES string of the molecule and ending with an approximation of static 3D charge distribution. It was thought that giving information on electrostatic potentials would give some information regarding the molecule dynamics within a biological system. However, it was found that adding the electrostatic potentials of each of the molecules severely harmed the accuracy and fidelity of the model. Further optimization of the data pipeline may aid the performance of the model.

We remain confident that it is feasible to construct a models using identified methods for interpretability which have not yet been tested in the computational toxicology space. We were able to compare the utility of both LIME and the graph convolution network explainer on the toxicology models, but further optimization of explanations of toxicological models is necessary to establish further trust in the models' predictions.

7 References

[1] Peter Bloomingdale, Conrad Housand, Joshua F. Apgar, et al. "Quantitative systems toxicology". In: *Current Opinion in Toxicology*. Translational Toxicology: Biomarkers 4 (June 2017), pp. 79–87. ISSN: 2468-2020. DOI: 10.1016/j.cotox.2017.07.003. URL: http://www.sciencedirect.com/science/article/pii/S2468202017300700 (visited on 05/24/2018).

[2] Robert J. Kavlock, Gerald Ankley, Jerry Blancato, et al. "Computational Toxicology—A State of the Science Mini Review". en. In: *Toxicological Sciences* 103.1 (May 2008), pp. 14–27. ISSN: 1096-6080. DOI: 10.1093/toxsci/kfm297. URL: https://academic.oup.com/toxsci/article/103/1/14/1693695 (visited on 05/24/2018).

[3] Luis G. Valerio. "In silico toxicology for the pharmaceutical sciences". In: *Toxicology and Applied Pharmacology* 241.3 (Dec. 2009), pp. 356–370. ISSN: 0041-008X. DOI: 10.1016/j.taap.2009.08.022. URL: http://www.sciencedirect.com/science/article/pii/S0041008X09003652 (visited on 05/24/2018).

[4] *CFR - Code of Federal Regulations Title 21*. URL: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=58 (visited on 04/02/2021).

[5] Gail A. Van Norman. "Drugs, Devices, and the FDA: Part 1: An Overview of Approval Processes for Drugs". In: *JACC: Basic to Translational Science* 1.3 (Apr. 2016), pp. 170–179. ISSN: 2452-302X. DOI: 10.1016/j.jacbts.2016.03.002. URL: http://www.sciencedirect.com/science/article/pii/S2452302X1600036X (visited on 10/09/2018).

[6] Joseph A. DiMasi, Henry G. Grabowski, and Ronald W. Hansen. "Innovation in the pharmaceutical industry: New estimates of R&D costs". en. In: *Journal of Health Economics* 47 (May 2016), pp. 20–33. ISSN: 01676296. DOI: 10.1016/j.jhealeco.2016.01.012. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167629616000291 (visited on 10/09/2018).

[7] Leah Isakov, Andrew W. Lo, and Vahid Montazerhodjat. *Is the FDA Too Conservative or Too Aggressive?: A Bayesian Decision Analysis of Clinical Trial Design*. en. SSRN Scholarly Paper ID 2641547. Rochester, NY: Social Science Research Network, Nov. 2017. URL: https://papers.ssrn.com/abstract=2641547 (visited on 10/09/2018).

[8] Frederick J Manning and Morton Swartz. "Committee to Review the Fialuridine (FIAU/FIAC) Clinical Trials Division of Health Sciences Policy". en. In: *Clinical Trials* (1995), p. 280.

[9] H Attarwala. "TGN1412: From Discovery to Disaster". In: *Journal of Young Pharmacists : JYP* 2.3 (2010), pp. 332–336. ISSN: 0975-1483. DOI: 10.4103/0975-1483.66810. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2964774/ (visited on 10/09/2018).

[10] Craig A. Umscheid, David J. Margolis, and Craig E. Grossman. “Key Concepts of Clinical Trials: A Narrative Review”. In: *Postgraduate Medicine* 123.5 (Sept. 2011), pp. 194–204. ISSN: 0032-5481. DOI: 10.3810/pgm.2011.09.2475. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3272827/ (visited on 05/24/2018).

[11] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, et al. “DeepTox: Toxicity Prediction using Deep Learning”. English. In: *Frontiers in Environmental Science* 3 (2016). ISSN: 2296-665X. DOI: 10.3389/fenvs.2015.00080. URL: https://www.frontiersin.org/articles/10.3389/fenvs.2015.00080/full (visited on 10/30/2018).

[12] William L. Hamilton, Rex Ying, and Jure Leskovec. “Representation Learning on Graphs: Methods and Applications”. In: *arXiv:1709.05584 [cs]* (Sept. 2017). arXiv: 1709.05584. URL: http://arxiv.org/abs/1709.05584 (visited on 12/11/2018).

[13] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. “Modeling polypharmacy side effects with graph convolutional networks”. In: *Bioinformatics* 34.13 (July 2018), pp. i457–i466. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty294. URL: https://doi.org/10.1093/bioinformatics/bty294 (visited on 04/01/2021).

[14] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, et al. “Recent advances in convolutional neural networks”. In: *Pattern Recognition* 77 (May 2018), pp. 354–377. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2017.10.013. URL: http://www.sciencedirect.com/science/article/pii/S0031320317304120 (visited on 09/20/2018).

[15] Wei Yu, Kuiyuan Yang, Hongxun Yao, et al. “Exploiting the complementary strengths of multi-layer CNN features for image retrieval”. In: *Neurocomputing* 237 (May 2017), pp. 235–241. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2016.12.002. URL: http://www.sciencedirect.com/science/article/pii/S0925231216314734 (visited on 09/20/2018).

[16] Joan Bruna, Wojciech Zaremba, Arthur Szlam, et al. “Spectral Networks and Locally Connected Networks on Graphs”. In: *arXiv:1312.6203 [cs]* (May 2014). arXiv: 1312.6203. URL: http://arxiv.org/abs/1312.6203 (visited on 04/02/2021).

[17] OECD. *Guidance Document on the Validation of (Quantitative) Structure-Activity Relationship [(Q)SAR] Models.* Type: doi:https://doi.org/10.1787/9789264085442-en. 2014. URL: https://www.oecd-ilibrary.org/content/publication/9789264085442-en.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “”Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *arXiv:1602.04938 [cs, stat]* (Aug. 2016). arXiv: 1602.04938. URL: http://arxiv.org/abs/1602.04938 (visited on 04/02/2021).

[19] Scott Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". en. In: (May 2017). URL: https://arxiv.org/abs/1705.07874v2 (visited on 04/02/2021).

[20] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Anchors: High Precision Model-Agnostic Explanations". en. In: (2018), p. 9.

[21] Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, et al. "Explainability Methods for Graph Convolutional Neural Networks". en. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* Long Beach, CA, USA: IEEE, June 2019, pp. 10764–10773. ISBN: 978-1-72813-293-8. DOI: 10.1109/CVPR.2019.01103. URL: https://ieeexplore.ieee.org/document/8954227/ (visited on 03/30/2021).

[22] David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *Journal of Chemical Information and Computer Sciences* 28.1 (Feb. 1988). Publisher: American Chemical Society, pp. 31–36. ISSN: 0095-2338. DOI: 10.1021/ci00057a005. URL: https://pubs.acs.org/doi/abs/10.1021/ci00057a005 (visited on 03/31/2021).

[23] Mario Krenn, Florian Häse, AkshatKumar Nigam, et al. "Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation". en. In: *Machine Learning: Science and Technology* 1.4 (Nov. 2020). Publisher: IOP Publishing, p. 045024. ISSN: 2632-2153. DOI: 10.1088/2632-2153/aba947. URL: https://doi.org/10.1088/2632-2153/aba947 (visited on 03/31/2021).

[24] David Weininger, Arthur Weininger, and Joseph L. Weininger. "SMILES. 2. Algorithm for generation of unique SMILES notation". In: *Journal of Chemical Information and Computer Sciences* 29.2 (May 1989). Publisher: American Chemical Society, pp. 97–101. ISSN: 0095-2338. DOI: 10.1021/ci00062a008. URL: https://doi.org/10.1021/ci00062a008 (visited on 03/31/2021).

[25] Noel M. O'Boyle. "Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI". In: *Journal of Cheminformatics* 4.1 (Sept. 2012), p. 22. ISSN: 1758-2946. DOI: 10.1186/1758-2946-4-22. URL: https://doi.org/10.1186/1758-2946-4-22 (visited on 03/31/2021).

[26] Laurianne David, Amol Thakkar, Rocío Mercado, et al. "Molecular representations in AI-driven drug discovery: a review and practical guide". In: *Journal of Cheminformatics* 12.1 (Sept. 2020), p. 56. ISSN: 1758-2946. DOI: 10.1186/s13321-020-00460-5. URL: https://doi.org/10.1186/s13321-020-00460-5 (visited on 03/31/2021).

[27] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, et al. "Molecular fingerprint similarity search in virtual screening". en. In: *Methods.* Virtual Screening 71 (Jan. 2015), pp. 58–63. ISSN: 1046-2023. DOI: 10.1016/j.ymeth.2014.08.005. URL: https://www.sciencedirect.com/science/article/pii/S1046202314002631 (visited on 03/31/2021).

[28] David Rogers and Mathew Hahn. "Extended-Connectivity Fingerprints". In: *Journal of Chemical Information and Modeling* 50.5 (May 2010). Publisher: American Chemical Society, pp. 742–754. ISSN: 1549-9596. DOI: 10.1021/ci100050t. URL: https://doi.org/10.1021/ci100050t (visited on 03/31/2021).

[29] Chihae Yang, Aleksey Tarkhov, Jörg Marusczyk, et al. "New Publicly Available Chemical Query Language, CSRML, To Support Chemotype Representations for Application to Data Mining and Modeling". en. In: *Journal of Chemical Information and Modeling* 55.3 (Mar. 2015), pp. 510–528. ISSN: 1549-9596, 1549-960X. DOI: 10.1021/ci500667v. URL: https://pubs.acs.org/doi/10.1021/ci500667v (visited on 04/02/2021).

[30] Malgorzata Natalia Drwal, Vishal Babu Siramshetty, Priyanka Banerjee, et al. "Molecular similarity-based predictions of the Tox21 screening outcome". English. In: *Frontiers in Environmental Science* 3 (2015). Publisher: Frontiers. ISSN: 2296-665X. DOI: 10.3389/fenvs.2015.00054. URL: https://www.frontiersin.org/articles/10.3389/fenvs.2015.00054/full (visited on 04/01/2021).

[31] Priyanka Banerjee, Frederic O. Dehnbostel, and Robert Preissner. "Prediction Is a Balancing Act: Importance of Sampling Methods to Balance Sensitivity and Specificity of Predictive Models Based on Imbalanced Chemical Data Sets". In: *Frontiers in Chemistry* 6 (Aug. 2018). ISSN: 2296-2646. DOI: 10.3389/fchem.2018.00362. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6149243/ (visited on 03/08/2019).

[32] Z. Wu et al. "MoleculeNet: a benchmark for molecular machine learning". English. In: *Royal Society of Chemistry* 9.N/A (Oct. 2017), pp. 513–530. ISSN: Could not find. DOI: 10.1039/C7SC02664A. URL: http://pubs.rsc.org/en/content/articlehtml/2018/sc/c7sc02664a.

[33] Ajay N. Jain and Anthony Nicholls. "Recommendations for evaluation of computational methods". In: *Journal of Computer-Aided Molecular Design* 22.3-4 (Mar. 2008), pp. 133–139. ISSN: 0920-654X. DOI: 10.1007/s10822-008-9196-5. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2311385/ (visited on 04/01/2021).

[34] Gabriel Idakwo, Sundar Thangapandian, Joseph Luttrell, et al. "Structure–activity relationship-based chemical classification of highly imbalanced Tox21 datasets". In: *Journal of Cheminformatics* 12.1 (Oct. 2020), p. 66. ISSN: 1758-2946. DOI: 10.1186/s13321-020-00468-x. URL: https://doi.org/10.1186/s13321-020-00468-x (visited on 04/01/2021).

[35] *deepchem/deepchem*. original-date: 2015-09-24T23:20:28Z. Apr. 2021. URL: https://github.com/deepchem/deepchem (visited on 04/01/2021).

[36] Noel M. O'Boyle, Michael Banck, Craig A. James, et al. "Open Babel: An open chemical toolbox". In: *Journal of Cheminformatics* 3.1 (Oct. 2011), p. 33. ISSN: 1758-2946. DOI: 10.1186/1758-2946-3-33. URL: https://doi.org/10.1186/1758-2946-3-33 (visited on 05/04/2021).

[37] Prakash Chandra Rathi, R. Frederick Ludlow, and Marcel L. Verdonk. "Practical High-Quality Electrostatic Potential Surfaces for Drug Discovery Using a Graph-Convolutional Deep Neural Network". eng. In: *Journal of Medicinal Chemistry* 63.16 (Aug. 2020), pp. 8778–8790. ISSN: 1520-4804. DOI: `10.1021/acs.jmedchem.9b01129`.

[38] Piers A. Townsend and Matthew N. Grayson. "Density Functional Theory in the Prediction of Mutagenicity: A Perspective". In: *Chemical Research in Toxicology* 34.2 (Feb. 2021). Publisher: American Chemical Society, pp. 179–188. ISSN: 0893-228X. DOI: `10.1021/acs.chemrestox.0c00113`. URL: `https://doi.org/10.1021/acs.chemrestox.0c00113` (visited on 04/01/2021).

8 Appendix A: Nomenclature

Terms/Abbreviations	**Definitions**
Black Box	A function that returns outputs given inputs without knowledge of its implementation.
Data Curation	Involves collecting and processing data: aggregating relevant data from a data set, removing duplicates and problematic chemical forms whose descriptors cannot be calculated.
Decision Tree	A tree-like model of decisions and consequences. Represents an algorithm that tests selected attributes to be classified.
Model Validation	Evaluation of model performance and deriving various metrics which will allow for comparison of the performance of the model.
Molecular Descriptor	An encoding of chemical information for mathematical treatment.
Multitask Learning	An approach to machine learning in which multiple tasks are learned simultaneously to exploit their commonalities.
Quantitative Structure-Activity Relationship (QSAR)	Classification model based on chemical structural similarities. Relates structure to biological activity.
Quantitative Structure-Property Relationship (QSPR)	Analogous to a QSAR; relates structure to a chemical property.
Random Forest	A machine learning method using many decision trees as base learners in an ensemble.
Structural Alert	Chemical substructure associated with a toxicity endpoint.
Toxicity Endpoint	Adverse physicochemical or biological effect defined with experimental conditions and protocols.
Training Set	Data set used as input for a model in the development process.

Classification Model	A model that has a finite number of discrete output states, called classes.
Validation Set	Data set used as input for a model to analyze its effectiveness. The results from the model's output are compared against experimental values.
Machine Learning Methods	Algorithms which converge to a model of the data by learning from a training set.