Discovering Molecular Functional Groups Using Graph Convolutional Neural Networks

Phillip E. Pope [†]*, Soheil Kolouri [†]*, Mohammad Rostrami*, Charles E. Martin*, Heiko Hoffmann*
* HRL Laboratories, LLC, Malibu, CA 91302
pepope@hrl.com, skolouri@hrl.com, mrostrami@hrl.com, cemartin@hrl.com, hhoffmann@hrl.com

Abstract

Functional groups (FGs) serve as a foundation for analyzing chemical properties of organic molecules. Automatic discovery of FGs will impact various fields of research, including medicinal chemistry, by reducing the amount of lab experiments required for discovery or synthesis of new molecules. Here, we investigate methods based on graph convolutional neural networks (GCNNs) for localizing FGs that contribute to specific chemical properties. Molecules are modeled as undirected graphs with atoms as nodes and bonds as edges. Using this graph structure, we trained GCNNs in a supervised way on experimentally-validated molecular training sets to predict specific chemical properties, e.g., toxicity. Upon learning a GCNN, we analyzed its activation patterns to automatically identify FGs using four different methods: gradient-based saliency maps, Class Activation Mapping (CAM), gradient-weighted CAM (Grad-CAM), and Excitation Back-Propagation. We evaluated the contrastive power of these methods with respect to the specificity of the identified molecular substructures and their relevance for chemical functions. Grad-CAM had the highest contrastive power and generated qualitatively the best FGs. This work paves the way for automatic analysis and design of new molecules.

I. Introduction

Automated design of molecules with desired properties is an emerging field with important applications including drug discovery [6] and material design [9]. Despite the complex structure of many molecules, in particular organic molecules, specific properties, e.g., toxicity or solubility in water, may be caused by substructures within a molecule. These substructures are called functional groups (FGs); e.g., toxophores are a specific type of FG's that produce toxicity in toxin molecules [21]. The study of common FGs is central to organic chemistry [3], see, e.g., the hydroxyl FG ($-OH$), as they can be used to systematically predict behavior of a compound in chemical reactions. Identifying of FGs can hint how to design molecules that posses particular properties.

The experimental approach to FG discovery requires high-precision instruments and many hours of expert supervision. In modern applications such as drug discovery, estimated to have a space of 10^{23} to 10^{60} molecules [19], experimentally searching the space is infeasible. Fortunately, recent efforts in combining computational chemistry and machine learning (ML) leverage data driven methods suggests possibility of efficiently narrowing this search space [5], [9], [27]. This stems from success of machine learning in areas such as computer vision and natural language processing.

Recent success in computer vision is due to emergence of deep convolutional neural networks (CNNs) [16] that has led to state of the art performance on classification [10] and object detection [20] tasks. The success of CNNs may be attributed to convolutional layers, which reduce the number of learnable parameters and allow deeper networks for multi-level abstraction of feature extraction. In addition, these features may be used to localize signal to regions of the input, giving a means of interpreting decisions by the network [29]. 'Despite this success, traditional deep CNNs are designed for Euclidean space where data is defined on a structured grid, e.g. domain of an image, as convolution is an operation defined on Euclidean space. For this reason, CNNs cannot be used directly on domains with different data structures such as graph structured data.

A recent variant of CNNs designed for graph structured data are graph convolution neural networks (GCNNs) [5], [8], which extend the definition of convolution to non-Euclidean spaces. GCNNs inherit ideas like shared weights and deep hierarchical feature distillation from CNNs and have led to promising results in classifying graph-structured

[†] denotes equal contribution

data [14]. Building upon the success of CNNs in computer vision, a recent line of research has applied GCNNs to atomic and molecular applications [5], [11], [23], [27], [9]. Inheriting properties of CNNs, GCNNs have led to promising results. In these methods, molecules are modeled as graphs, where the graph nodes represent the atoms, and the graph edges (potentially weighted) represent the chemical bonds and their types, and learning is performed on the molecule-level.

Here, we investigate, for the first time, GCNN methods that determine localized parts of a graph corresponding to a specific classification, as inspired by related work on images [29]. We compare four different methods: gradient-based saliency maps, Class Activation Mapping (CAM), Gradient-weighted Class Activation Mapping (Grad-CAM), and Excitation Back-Propagation (EBP). We evaluate their performance with respect to contrastiveness (class-specific localization) and sparsity of localization and qualitatively compare the localization heatmaps over the graph structure. For the graph input, we focus on molecules. So, highlighted structural components can show FGs that correspond to a certain chemical property.

II. Related Work

Various ML techniques have been recently applied to applications within molecular level chemistry and biochemistry. Given a proper feature extraction method that can convert all molecules in a dataset to fixed-size vectors, we can use ML techniques to automatically predict chemical characteristics of molecules [4]. But finding the right feature extraction method is difficult and can be restrictive. Recent ML progress is mainly based on the emergence of deep neural architectures including CNNs, which automatize the process of feature extraction in an end-to-end and data driven scheme. The challenge of applying CNNs on molecule datasets is that CNNs can only receive structured data as their input, e.g., images. This limitation has been circumvented by the invention of graph convolutional neural networks (GCNN). GCNNs provide an extension of CNNs to non-Euclidean spaces and are suitable for handling graph structured data such as molecules. Similar to CNNs, GCNNs are able to learn descriptive features automatically that outperform engineered features, enabling GCNNs to achieve state-of-the-art performance on several chemical prediction tasks, including toxicity prediction [11], solubility [5], and energy prediction [23].

A long standing limitation of general deep neural networks has been the difficulty in interpreting and explaining the classification results. Recently, explainability methods have been devised for deep networks and specifically CNNs [25], [29], [24], [28]. These methods enable one to probe a CNN and identify the important substructures of the input data (as deemed by the network), which could be used as an explanatory tool or as a tool to discover unknown underlying substructures in the data. For example, in the area of medical imaging, in addition to classifying images having malignant lesions, they can be localized, as the CNN can provide reasoning for classifying an input image. Here, we are interested in measuring the potential of these methods for discovery of FGs in organic molecules.

The most straight-forward approach for generating a sensitivity map over the input data to discover the importance of the underlying substructures is to calculate a gradient map within a layer by considering the norm of the gradient vector with respect to an input for each network weight [25]. However, gradient maps are known to be noisy and smoothening these maps might be necessary [26]. More advanced techniques include Class Activation Mapping (CAM) [29], Gradient-weighted Class Activation Mapping (Grad-CAM) [24], and Excitation Back-Propagation (EB) [28] techniques that improve gradient maps by taking into account some notion of context. These techniques have been shown to be effective on CNNs and can identify highly abstract notions in images.

Inspired by the explainability power of deep CNNs, our goal is to adapt these techniques for deep GCNNs to automatize discovery of chemical FGs for a particular behavior. This process can be particularly helpful for FGs because, as opposed to images, humans cannot intuitively determine the relevant context within a molecule for a particular property of that molecule. Our specific contributions in this work are:

- Adapting explanation tools for CNNs to GCNNs with the application of discovering FGs for organic molecules,
- Comparing the contrastive power and class specificity of the explainability methods in identifying FGs, and
- Analyzing three molecular datasets and their identified FGs.

We envision that our proposed framework could help chemists with identifying new FGs that have not been discovered before, reducing the experimental cost and the required time needed for this purpose.

III. Methods

We compare and contrast the application of four popular explainability methods to Graph Convolutional Neural Networks (GCNNs). These methods are gradient-based saliency maps [25], Class Activation Mapping (CAM)

[29], Gradient-weighted Class Activation Mapping (Grad-CAM) [24], and Excitation Back-Propagation (EBP) [28]. Furthermore, we explore the benefits of a number of enhancements to these approaches.

A. Explainability for Convolutional Neural Networks

Perhaps the most straight-forward (and well-established) approach is that of gradient-based saliency maps [25]. In this approach, one simply differentiates the output of the model with respect to the model input, thus creating a heat-map, through the norm of the gradient over input variables, indicating their relative importance. Note that the resulting gradient in the input space points in the direction corresponding to the maximum positive rate of change in the model output. Therefore the negative values in the gradient are discarded to only retain the parts of input that positively contribute to the solution:

$$L^c_{Gradient} = \|\text{ReLU}\left(\frac{\partial y^c}{\partial x}\right)\| \quad , \tag{1}$$

where y^c is the score for class c before the softmax layer, and x is the input. While easy to compute and interpret, saliency maps generally perform worse than newer techniques (like CAM, Grad-CAM, and EB), and it was recently argued that saliency maps tend to represent noise rather than signal [13].

The CAM approach provides an improvement over saliency maps for convolutional neural networks, including GCNNs, by identifying important, class-specific features at the last convolutional layer as opposed to the input space. It is well-known that such features tend to be more semantically meaningful (e.g., faces instead of edges). The downside of CAM is that it requires the layer immediately before the softmax classifier (output layer) to be a convolutional layer followed by a global average pooling (GAP) layer. This precludes the use of more complex, heterogeneous networks, such as those that incorporate several fully connected layers before the softmax layer.

To compute CAM, let $F_k \in \mathbb{R}^{u \times v}$ be the k^{th} feature map of the convolutional layer preceding the softmax layer. Denote the global average pool (GAP) of F_k by

$$e_k = \frac{1}{Z}\sum_i \sum_j F_{k,i,j} \tag{2}$$

where $Z = uv$. Then, a given class score, y^c, can be defined as

$$y^c = \sum_k w^c_k e_k, \tag{3}$$

where the weights $\mathbf{w}^c_k$ are learned based on the input-output behavior of the network. The weight w^c_k encodes the importance of feature k for predicting class c. By upscaling each feature map to the size of the input images (to undo the effect of pooling layers) the class-specific heat-map in the pixel-space becomes

$$L^c_{CAM}[i,j] = \text{ReLU}\left(\sum_k w^c_k F_{k,i,j}\right). \tag{4}$$

The Grad-CAM method improves upon CAM by relaxing the architectural restriction that the penultimate layer must be a convolutional. This is achieved by using feature map weights α^c_k that are based on back-propagated gradients. Specifically, Grad-CAM defines the weights according to

$$\alpha^c_k = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial F_{k,i,j}}. \tag{5}$$

Following the intuition behind Equation (4) for CAM, the heat-map in the pixel-space according to Grad-CAM is computed as

$$L^c_{Grad-CAM}[i,j] = \text{ReLU}\left(\sum_k \alpha^c_k F_{k,i,j}\right), \tag{6}$$

where the ReLU function ensures that only features that have a *positive* influence on the class prediction are non-zero.

Excitation Back-Propagation is an intuitively simple, but empirically effective explanation method. In [22], it is argued and demonstrated experimentally that explainability approaches such as EB [28], which ignore nonlinearities in the backward-pass through the network, are able to generate heat-maps that "conserve" evidence for or against a network predicting any particular class. Let a_i^l be the i'th neuron in layer l of a neural network and $a_j^{(l-1)}$ be a neuron in layer $(l-1)$. Define the *relative* influence of neuron $a_j^{(l-1)}$ on the activation $y_i^l \in \mathbb{R}$ of neuron a_i^l, where $y_i^l = \sigma(\sum_{ji} W_{ji}^{l-1} y_j^{(l-1)})$ and for $W^{(l-1)}$ being the synaptic weights between layers $(l-1)$ and l, as a probability distribution $P(a_j^{(l-1)})$ over neurons in layer $(l-1)$. This probability distribution can be factored as

$$P(a_j^{(l-1)}) = \sum_i P(a_j^{(l-1)}|a_i^l)P(a_i^l). \tag{7}$$

Zhang et al. then define the conditional probability $P(a_j^{(l-1)}|a_i^l)$ as

$$P(a_j^{(l-1)}|a_i^l) = \begin{cases} Z_i^{(l-1)} y_j^{(l-1)} W_{ji}^{(l-1)} & \text{if } W_{ji}^{(l-1)} \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where

$$Z_i^{(l-1)} = \left(\sum_j y_j^{(l-1)} W_{ji}^{(l-1)} \right)^{-1}$$

is a normalization factor such that $\sum_j P(a_j^{(l-1)}|a_i^l) = 1$. For a given input (e.g., an image), EB generates a heat-map in the pixel-space w.r.t. class c by starting with $P(a_i^L = c) = 1$ at the output layer and applying Equation (7) recursively.

These reviewed explainability methods are originally designed for CNNs, which are defined on a signal supported on a uniform grid. Here, we are interested in signals supported on non-Euclidean structures, e.g. graphs. In what follows, we first briefly discuss GCNNs and then describe the extensions of these explainability methods to GCNNs.

B. Graph Convolutional Neural Networks

Let an attributed graph with N nodes be defined with its node attributes $X \in \mathbb{R}^{N \times d_{in}}$ and its adjacency matrix $A \in \mathbb{R}^{N \times N}$ (weighted or binary). In addition, let the degree matrix for this graph be $D_{ii} = \sum_j A_{ij}$. Following the work of Kipf and Welling [15], we define the graph convolutional layer to be

$$F^l(X,A) = \sigma(\underbrace{\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}}_{V} F^{(l-1)}(X,A)W^l)\,, \tag{9}$$

where F^l is the convolutional activations at the $l'th$ layer, $F^0 = X$, $\tilde{A} = A + I_N$ is the adjacency matrix with added self connections where $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, $W^l \in \mathbb{R}^{d_l \times d_{l+1}}$ are the trainable convolutional weights, and $\sigma(\cdot)$ is the element-wise nonlinear activation function. Figure 1 shows the used GCNN architecture in this work, where the activations in layers $l = 1, 2, 3$ follow Eq. (9), which is a first-order approximation of localized spectral filters on graphs.

For molecule classification, each molecule can be represented as an attributed graph $\mathcal{G}_i = (X_i, A_i)$, where the node features X_i summarize the local chemical environment of the atoms in the molecule, including atom-types, hybridization types, and valence structures [27], and the adjacency matrix encodes atomic bonds and demonstrate the connectivity of the whole molecule (see Figure 1). For a given dataset of labeled molecules $\mathcal{D} = \{\mathcal{G}_i = (X_i, A_i), y_i\}_{i=1}^M$ with labels y_i indicating a certain chemical property, e.g., blood-brain-barrier penetrability or toxicity, the task is to learn a classifier that maps each molecule to its corresponding label, $g : (X_i, A_i) \rightarrow y_i$.

Given that our task is to classify individual graphs (i.e., molecules) with potentially different number of nodes, we use several layers of graph convolutional layers followed by a global average pooling (GAP) layer over the graph nodes (i.e. atoms). In this case, all graphs will be represented with a fixed size vector. Finally, the GAP features are fed to a classifier. To enable applicability of CAM [29], we simply used a softmax classifier after the GAP layer.

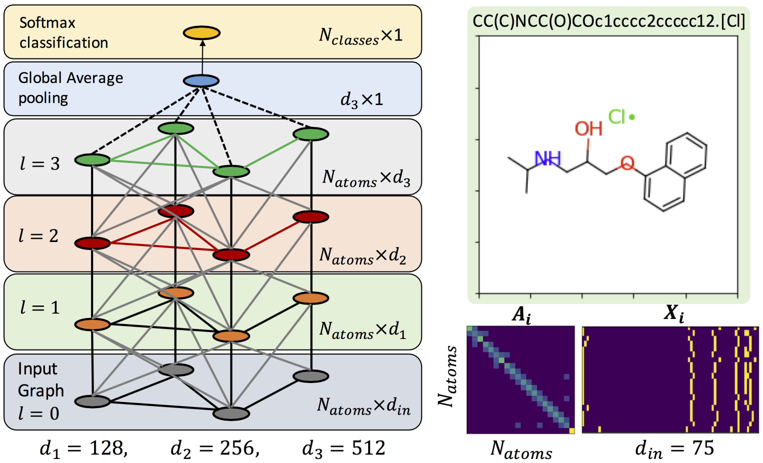

Fig. 1: Our GCNN architecture together with the visualization of the input feature and adjacency matrix for a sample molecule from BBBP dataset.

C. Explainability for Graph Convolutional Neural Networks

In this subsection, we describe the extension of CNN explainability methods to GCNNs. Let the k'th graph convolutional feature map at layer l be defined as:

$$F_k^l(X,A) = \sigma(VF^{(l-1)}(X,A)W_k^l) \tag{10}$$

where W_k^l denotes the $k'th$ column of matrix W^l, and $V = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ (see Eq. (8)). In this notation, for the n'th atom of the molecule, the k'th feature at the l'th layer is denoted by $F_{k,n}^l$. Then, the GAP feature after the final convolutional layer, L, is calculated as

$$e_k = \frac{1}{N}\sum_{n=1}^{N} F_{k,n}^L(X,A) \quad , \tag{11}$$

and the class score is calculated as, $y^c = \sum_k w_k^c e_k$. Using this notation, the explainability methods could be extended to GCNNs as follows:

Gradient-based atomic heat-maps for the n'th atom are calculated as

$$L_{Gradient}^c[n] = \|\text{ReLU}\left(\frac{\partial y^c}{\partial X_n}\right)\| \quad ; \tag{12}$$

CAM atomic heat-maps for the n'th atom are calculated as

$$L_{CAM}^c[n] = \text{ReLU}(\sum_k w_k^c F_{k,n}^L(X,A))) \,. \tag{13}$$

Grad-CAM's class specific weights for class c at layer l and for the k'th feature are calculated by

$$\alpha_k^{l,c} = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial y^c}{\partial F_{k,n}^l} \quad , \tag{14}$$

and the heat-map for the n'th atom calculated from layer l is

$$L_{Grad-CAM}^c[l,n] = \text{ReLU}(\sum_k \alpha_k^{l,c} F_{k,n}^l(X,A)) \,. \tag{15}$$

Grad-CAM enables us to generate heat-maps with respect to different layers of the network. In addition, for our model shown in Figure 1, Grad-CAM's heat-map at the final convolutional layer and CAM's heat-map are equivalent $L_{Grad-CAM}^c[L,n] = L_{CAM}^c[n]$ (See [24] for more details). In this work, we report results for $L_{Grad-CAM}^c[L,n]$ as well as

$$L_{Grad-CAM\,Avg}^c[n] = \frac{1}{L}\sum_{l=1}^{L} L_{Grad-CAM}^c[l,n] \,. \tag{16}$$

Excitation Backpropagation's heat-map for our model is calculated via backward passes through the softmax classifier, the GAP layer, and several graph convolutional layers. The equations for backward passes through the softmax classifier and the GAP layer are

$$\begin{cases} p(e_k) = \sum_c \frac{e_k ReLU(w_k^c)}{\sum_k e_k ReLU(w_k^c)} p(c) & \text{Softmax} \\ \\ p(F^L_{k,n}) = \frac{F^L_{k,n}}{Ne^k} p(e^k) & \text{GAP}\,, \end{cases} \tag{17}$$

where $p(c) = 1$ for the class of interest and zero otherwise. The backward passes through the graph convolutional layers, however, are more complicated. For notational simplicity, we decompose a graph convolutional operator into

$$\begin{cases} \hat{F}^l_{k,n} = \sum_m V_{n,m} F^l_{k,m} \\ \\ F^{(l+1)}_{k',n} = \sigma(\sum_{k'} \hat{F}^l_{k,n} W^l_{k,k'})\,, \end{cases} \tag{18}$$

where the first equation is a local averaging of atoms (with $V_{n,m} \geq 0$), and the second equation is a fixed perceptron applied to each atom (analogous to one-by-one convolutions in CNNs). The corresponding backward passes for these two functions can be defined as

$$\begin{cases} p(F^l_{k,n}) = \sum_m \frac{V_{n,m} F^l_{k,n}}{\sum_n V_{n,m} F^l_{k,m}} p(\hat{F}^l_{k,m}) \\ \\ p(\hat{F}^l_{k,n}) = \sum_{k'} \frac{\hat{F}^l_{k,n} ReLU(W^l_{k,k'})}{\sum_k \hat{F}^l_{k,n} ReLU(W^l_{k,k'})} p(F^{(l+1)}_{k',n})\,. \end{cases} \tag{19}$$

We generate the heat-map over the input layer by recursively backpropagating through the network and averaging the backpropagated probability heat-maps on the input:

$$L^c_{EB}[n] = \frac{1}{d_{in}} \sum_{k=1}^{d_{in}} p(F^0_{k,n}) \quad . \tag{20}$$

The contrastive extension of L^c_{EB} follows Eq. (8) in [28]; we call this contrastive variant, c-EB.

IV. Experiments

This section describes the experimental setup, results of class-specific explanations, and a substructure frequency analysis identifying relevant FGs.

A. Experimental Setup

We evaluated explanation methods on three binary classification molecular datasets, BBBP, BACE, and task NR-ER from TOX21 [27]. Each dataset contains binary classifications of small organic molecules as determined by experiment. The BBBP dataset contains measurements on whether a molecule permeates the human blood brain barrier and is of significant interest to drug design. The BACE dataset contains measurements on whether a molecule inhibits the human enzyme β-secretase. The TOX21 dataset contains measurements of molecules for several toxicity targets. We selected the NR-ER task from this data, which is concerned with activation of the estrogen receptor [18]. These datasets are imbalanced. Class ratios for each dataset are reported in Table I.

In addition, we followed the recommendations in [27] for train/validation/test partitioning, where random partitioning is not recommended due to domain-specific considerations. In particular, for BACE and BBBP, a ”scaffold” split is recommended, which partitions molecules according to their structure, i.e. structurally similar molecules are partitioned in the same split. As this is recommended in existing literature, we follow this convention.

Using 80:10:10 train/validation/test split, we report AUC-ROC and AUC-PR values of our trained model for each dataset in Table II. These results are comparable to those reported in [27], and confirm that the model was trained correctly.

For all datasets, we used the GCNN + GAP architecture as described in Figure 1 with the following configuration: three graph convolutional layers of size 128, 256, and 512, respectively, followed by a GAP layer, and a softmax classifier. Models were trained for 100 epochs using the ADAM optimizer with learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The models were implemented in Keras with Tensorflow backend [2].

	Positives	Negatives
BBBP	1560	479
BACE	691	821
TOX21	793	5399

TABLE I: Dataset class breakdown

	AUC-ROC			AUC-PR		
	Train	Validation	Test	Train	Validation	Test
BBBP	0.991 ± 0.001	0.991 ± 0.003	0.966 ± 0.007	0.994 ± 0.003	0.992 ± 0.007	0.966 ± 0.012
BACE	0.991 ± 0.001	0.986 ± 0.007	0.999 ± 0.001	0.939 ± 0.043	0.924 ± 0.036	0.999 ± 0.001
TOX21	0.868 ± 0.006	0.873 ± 0.009	0.881 ± 0.008	0.352 ± 0.047	0.348 ± 0.057	0.347 ± 0.048

TABLE II: Evaluation results for splits train/validation/test for each dataset. Unusually high test metrics are observed for BACE (both AUC-ROC and AUC-PR), and TOX21 (AUC-ROC). Despite the surprising outcome, our result is consistent with results in (Wu et al. 2018, Figure 7, bottom right). The scaffold split is believed to be responsible.

B. *Class-Specific Explanations*

After training models for each dataset, we computed each explanation method on all samples and generated heap-maps showing relevant substructures (Figure 3 shows selected results). The heat-maps are calculated for positive and negative classes and normalized for each molecule across both classes and nodes to form a probability distribution over the input nodes. Class specificity can be seen by comparing explanations across classes within a method, i.e., nodes activated by one class tend to be inactivate for the other.

To quantitatively measure class specificity, we propose a measure based on the hamming distance between binarized heat-maps for positive and negative classes (Figure 2). We report the ratio of the hamming distance of the binarized heat-maps to the total number of identified atoms by heat-maps in Table III. Grad-CAM and c-EB showed the most contrastive power. In addition, to account for varying number of atoms and sparsity of identified substructures in different molecules, we include a measure of sparsity of the activation (Table IV). We define this measure as the number of identified atoms in *either* explanation, divided by the total number of atoms. The contrastive Excitation Back-Propagation method showed the sparsest activations.

C. *Substructure Frequency Analysis*

Analyzing a collection of molecules with a given property for common substructures is a known technique for discovering relevant functional groups [1], [17]. As the number of all possible substructures in a set of molecules is huge, these methods typically restrict analysis to a set of substructures obtained from a fragmentation algorithm. Here, we close the loop for discovering FGs by identifying functional molecular substructures from the machine learned generated heat-maps.

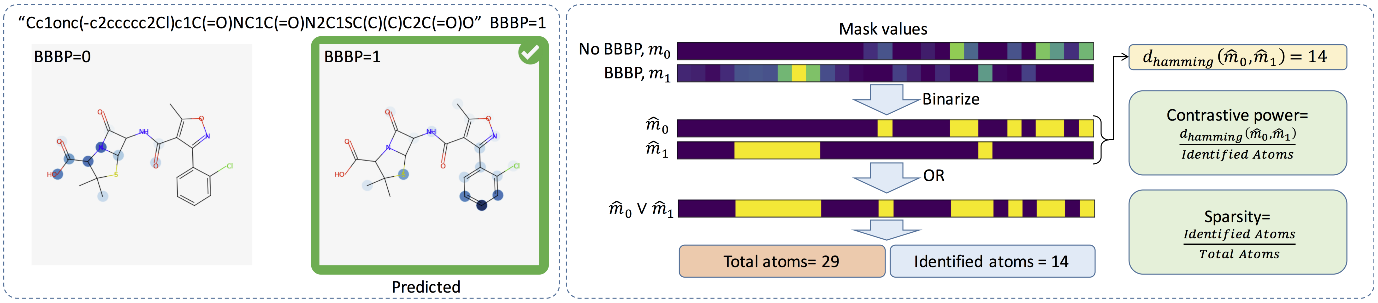
"Cc1onc(-c2ccccc2Cl)c1C(=O)NC1C(=O)N2C1SC(C)(C)C2C(=O)O" BBBP=1
BBBP=0
BBBP=1
Predicted
Mask values
No BBBP, m_0
BBBP, m_1
Binarize
$\hat{m}_0$
$\hat{m}_1$
OR
$\hat{m}_0 \vee \hat{m}_1$
Total atoms= 29
Identified atoms = 14
$d_{hamming}(\hat{m}_0, \hat{m}_1) = 14$
Contrastive power= $\frac{d_{hamming}(\hat{m}_0, \hat{m}_1)}{Identified\ Atoms}$
Sparsity= $\frac{Identified\ Atoms}{Total\ Atoms}$

Fig. 2: Visualization of the molecule "Oprea1_495871" with molecular formula "C19H17ClN3O5S", its corresponding SMILES representation, and the result of applying CAM to identify atoms that contribute to its BBBP characteristic (on the left), and the process of measuring contrastive power and sparsity of the method (i.e., CAM) for this molecule.

	BBBP	BACE	TOX21
CG	0.45± 2.19	0.77± 2.99	0.2 ± 2.13
Grad-CAM	99.99± 0.11	100.0± 0.0	99.99± 0.29
Grad-CAM Avg	41.06±19.05	29.22±14.04	44.03±23.7
EB	50.87±18.76	60.29±15.40	49.06±22.59
c-EB	96.97± 5.68	97.04±5.12	97.23±9.3

TABLE III: Our measure of contrastive power for different methods. The best performing methods are highlighted in green.

GCNN explanations (i.e., heat-maps) often occur on co-located regions in a molecule and provide a data driven means of generating candidate substructures. Further, we can analyze generated heat-maps for reoccurring patterns which may yield fruitful insights or help select candidate functional substructures. We devised an automated method for counting the occurrence of substructures identified from the heat-maps generated by the explanation methods. In short, we counted the frequency of each substructure observed in explanations for the three dataset and further analyzed their class specificity.

To identify each substructure, we took the largest connected components consisting of atoms with explanation value greater than some threshold (here, 0), which we call activated atoms, and edges between such atoms. After extracting the activated connected components as identified by the heat-maps, we count their frequency. This analysis requires comparing molecular substructures, a functionality found in open source computational chemistry libraries such as RDKit. We restricted our method to consider only exact substructure comparisons.

A prevalent substructure in the dataset may artificially show a high prevalence in the generated heat-maps. To account for this potential imbalance, we counted the occurrences of explanation-identified substructures in both positive and negative labeled data in the dataset. We used these counts to normalize the counts obtained from the explanations and construct two ratios:

$$R_e^s = \frac{N_e^s}{N_p^s + N_n^s}$$

and

$$R_p^s = \frac{N_p^s}{N_p^s + N_n^s}$$

where N_e^s, N_p^s and N_n^s are the number of times a substructure s occurs in explanations, the positively labeled data, and the negatively labeled data respectively. The ratio R_e^s measures how often a substructure occurs in explanations. The second one measures how often a substructure occurs in positively labeled data, and serves as a baseline for the first. Note that a high R_p^s corresponds to high class specificity for an identified substructure.

These ratios are sensitive to rare substructures. For instance if a substructure occurs only once and occurs in the explanations then it has $R_e^s = 1$. To mitigate this sensitivity, we report only substructures that occur more than 10 times in the dataset.

Figure 4 shows the most prominent substructures according to our analysis. Here, due to space limitation, we used only Grad-CAM to extract the explanations because it was the most contrastive method (Table III). Additionally, we restricted the explanations to those samples which were true positives. We ranked substructures by R_e^s and report the top 10 for each dataset. As shown in the Figure, the identified substructures have high class specificity. In addition, we observed a few patterns of known FGs being discovered by our method: Halogens (Cl,F,Br) are prevalent in explanations for BBBP, Amides are prevalent in explanations for BACE, and aromatic ring structures are prevalent for TOX21.

V. Conclusion

In this work, we extended explainability methods, which were designed for CNNs, to GCNNs, compared these methods qualitatively and quantitatively, and used them as a tool for discovering functional groups in organic molecules.

We compared four methods for identifying molecular substructures that are responsible for a given classification. The GradCAM methods could qualitatively identify functional groups that are known to be chemically active for

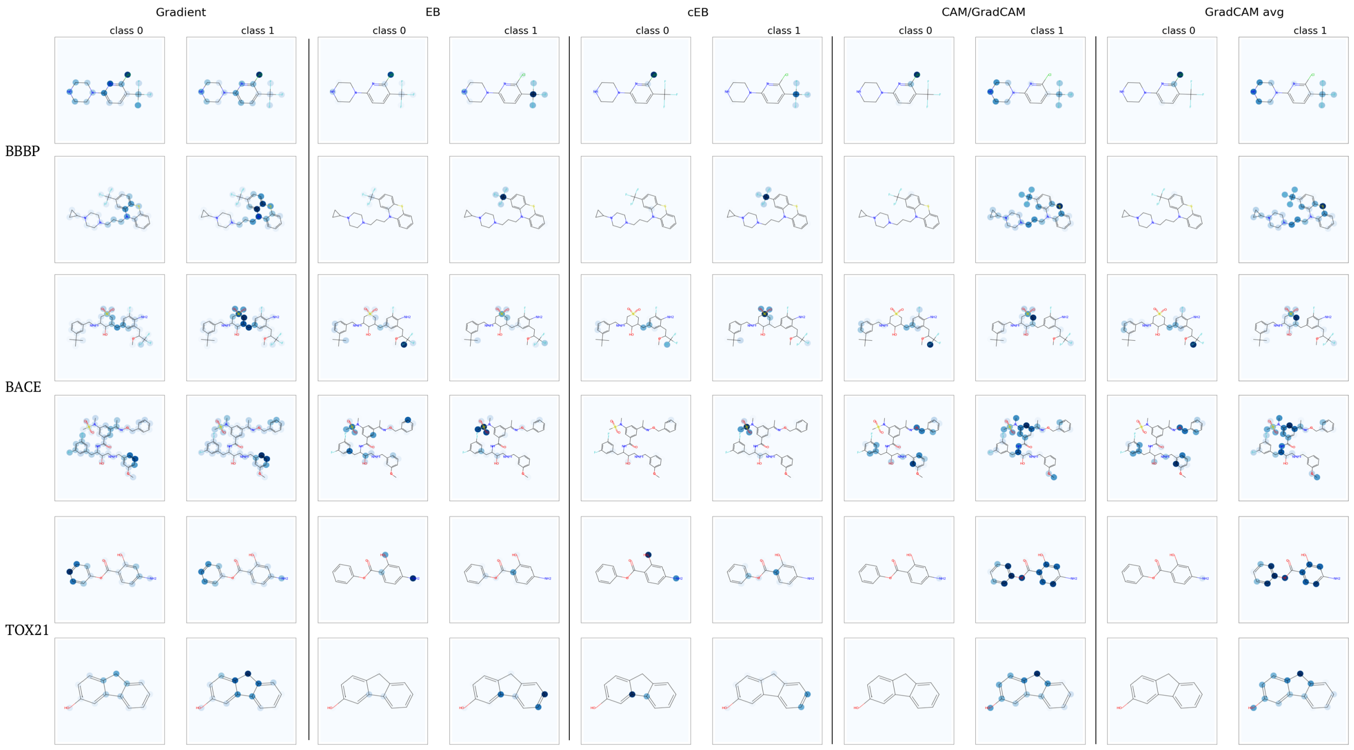

Fig. 3: Selected explanation results for each dataset, e.g., EB highlights CF3 for BBBP. Each sample is a true positive. Class specificity can be seen by comparing regions across classes for each method. A darker blue color indicates a higher relevance for a given class. The results for CAM and Grad-CAM are identical (see Methods). Best viewed on a computer screen.

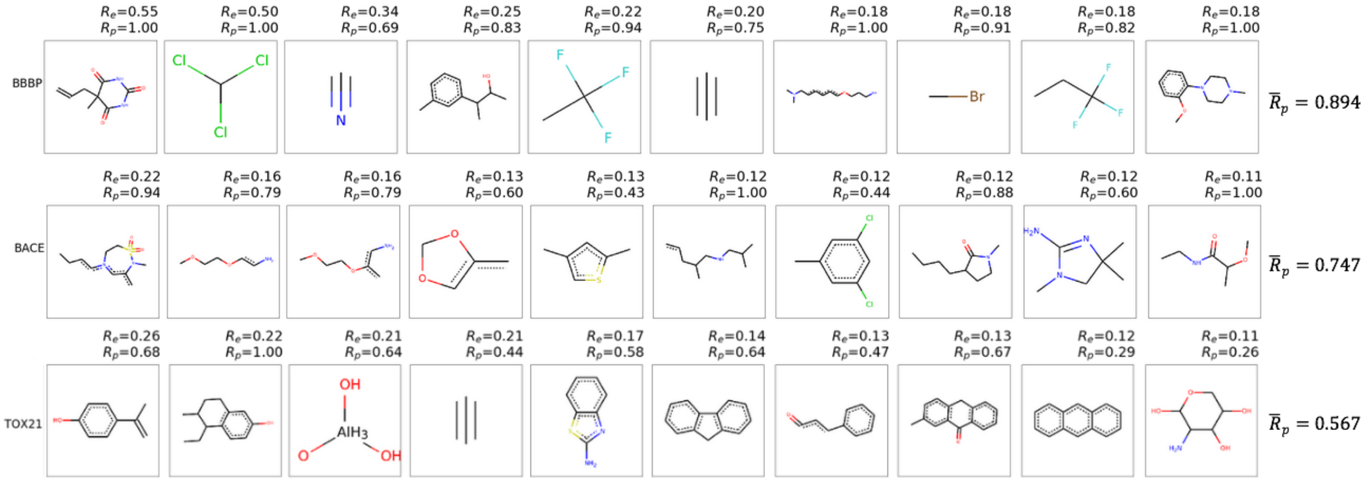

Fig. 4: Top 10 most prevalent substructures by dataset. We rank substructures by the ratio R_e, the number of times a substructure occurs in explanations over total occurrences in the dataset. For comparison, we also report the ratio R_p of how many times a substructure occurs in the positively labeled set over total occurrences. To account for rare structures, we report only substructures that occurred more than 10 times in the dataset. The right-most column shows average R_p values.

	BBBP	BACE	TOX21
CG	99.78± 2.43	99.72±1.58	99.79±2.98
Grad-CAM	93.74±7.83	90.64±7.67	95.14±8.74
Grad-CAM Avg	99.99±0.07	100.0±0.0	99.99±0.11
EB	59.65±22.11	48.6±13.97	69.88±23.04
c-EB	59.46±21.69	46.99±13.95	69.69±22.91

TABLE IV: Our measure of sparsity of identified atoms, lower values indicate higher sparsity of the identified atoms. The method with highest sparsity is highlighted in green.

the specified tasks, e.g., CF3 for brain blood-barrier penetration [7], [12]. For other identified functional groups, additional experiments are needed to confirm their chemical properties.

With our metrics, Grad-CAM and c-EB showed the best contrastive power for showing substructures for different classes. Compared to Grad-CAM, c-EB showed sparser activations, which could be an advantage in certain applications. For the benzene group, however, c-EB could not capture the entire group because the activation was too sparse. Here, GradCAM performed better. So, apparently, there is an optimal value for the sparsity, which may depend on the application.

Our results provide a pathway for automated discovery of functional relevant groups of molecules. Such an system is a stepping stone towards fully automated drug discovery. Finally, the proposed framework is a generic tool for discovering functional substructures in general graphs, including social networks and electrical grids.

References

[1] Y. Chen, F. Cheng, L. Sun, W. Li, G. Liu, and Y. Tang. Computational models to predict endocrine-disrupting chemical binding with androgen or oestrogen receptors. *Ecotoxicology and Environmental Safety*, 110:280 – 287, 2014.

[2] F. Chollet. keras. https://github.com/fchollet/keras, 2015.

[3] F. E. Critchfield. Organic functional group analysis. 1963.

[4] G. E. Dahl, N. Jaitly, and R. Salakhutdinov. Multi-task neural networks for qsar predictions. *arXiv preprint arXiv:1406.1231*, 2014.

[5] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[6] E. Gawehn, J. A. Hiss, and G. Schneider. Deep learning in drug discovery. *Molecular informatics*, 35(1):3–14, 2016.

[7] A. Ghosh, M. Brindisi, and T. J. Developing b-secretase inhibitors for treatment of alzheimers disease. *Journal of Neurochemistry*, 120:71–83, 2012.

[8] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, 2017.

[9] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[11] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[12] S. Kim, H. Lee, and I. Kim. Robust neuroprotective effects of 2-((2-oxopropanoyl)oxy)-4-(trifluoromethyl)benzoic acid (optba), a htb/pyruvate ester, in the postischemic rat brain. *Scientific Reports*, 6, 2016.

[13] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. The (un) reliability of saliency methods. *arXiv preprint arXiv:1711.00867*, 2017.

[14] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[15] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Advances in neural information processing systems*, 2017.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[17] A. Lombardo, F. Pizzo, E. Benfenati, A. Manganaro, T. Ferrari, and G. Gini. A new in silico classification model for ready biodegradability, based on molecular fragments. *Chemosphere*, 108:10 – 16, 2014.

[18] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter. Deeptox: toxicity prediction using deep learning. *Frontiers in Environmental Science*, 3:80, 2016.

[19] P. G. Polishchuk, T. I. Madzhidov, and A. Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, 27(8):675–679, 2013.

[20] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[21] J. E. Ridings, M. D. Barratt, R. Cary, C. G. Earnshaw, C. E. Eggington, M. K. Ellis, P. N. Judson, J. J. Langowski, C. A. Marchant, M. P. Payne, et al. Computer prediction of possible toxic action from chemical structure: an update on the derek system. *Toxicology*, 106(1-3):267–279, 1996.

[22] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2017.

[23] K. Schütt, P. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko, and K. Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, pages 991–1001, 2017.

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017.

[25] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[26] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[27] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[28] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision*, pages 543–559. Springer, 2016.

[29] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.