SAPIENZA
UNIVERSITÀ DI ROMA

# Explainable AI in Drug Design: Perturbation based molecular attributions using Graph Convolutional Networks


Faculty of Information Engineering, Informatics, and Statistics

Department of Computer, Control and Management Engineering "Antonio Ruberti"

Master's Degree in Artificial Intelligence and Robotics

Candidate

Alessio Ragno

ID number 1759198

Thesis Advisor

Prof. Roberto Capobianco



Academic Year 2020/2021

Thesis not yet defended

:

Reviewer: Prof. Aris Anagnostopoulos

---

**Explainable AI in Drug Design: Perturbation based molecular attributions using Graph Convolutional Networks**

Master's thesis. Sapienza – University of Rome



This thesis has been typeset by LaTeX and the Sapthesis class.

Version: May 19, 2021

Author's email: ragno.1759198@studenti.uniroma1.it

*Dedicated to*
*My Family, my girlfriend and my friends*
*who have been by my side during this journey*

# Abstract

Since the introduction of big data in chemistry has emerged the necessity to dissect how molecular property variation is modulated by either single atoms or chemical groups. Herein, it is proposed a perturbation based approach to train Graph-to-Graph Neural Network (GGN) to the field of medicinal chemistry. As initial case studies the method was applied to solubility and molecular acidity while checking its consistency in comparison with known experimental chemical data. As final goal, the GGN approach could represent a valuable medicinal chemistry tool aimed to tackle unsolved or hurdle problems with particular regards to activity cliffs, lead optimization and de-novo drug design.

# Contents

# Chapter 1

# Introduction

Artificial Intelligence (AI) brought up big improvements in a variety of scientific subjects such as physics and chemistry. Neural networks (NN) turned out to be a powerful tool for extracting multiple level features from a given dataset. With the improvement of computers' calculation power, data scientists succeeded in managing huge NN architectures to solve complicated tasks such as image object detection and generation. The scaling up of these architectures increased the performance of NN models over multiple tasks. On the other hand as more features are used, it usually becomes difficult to understand their inner functioning and therefore models' interpretation.

Explainable AI (XAI) aims to generate Artificial Intelligence models that can be interpreted in different ways. One major approach of this field is the generation of attributions of the input with respect to the output, which is, assigning scores to different parts of the input to find out salient parts that contribute to the property under prediction. One popular example of this approach is the usage of XAI to attributing pixels in order to highlight salient parts of images (Shrikumar et al. [2017], Zhou et al. [2015], Bach et al. [2015], Sundararajan et al. [2017]). The outputs of this approach are generally called saliency maps.

A similar drawback is also faced in the field of computational medicinal chemistry (Med Chem). A common task in Med Chem, and more specifically in ligand based drug design (LBDD), relies in defining relationships between bio-active molecule structures and their biological properties, which is, the so called QSAR (Quantitative Structure-Activity Relationships) or more generally QSPR (Quantitative Structure-Property Relationships). Medicinal chemists generally use machine learning models for predicting the activity of virtualized molecules in order to design new potentially active molecules by tweaking the structure of already existing ones. XAI could contribute to this field with the generation of saliency maps that can highlight structural moieties of the molecules which tend to increase the activity and those parts which tend to impair it.

This study proposes a new method for tackling this problem using graph convolutional

neural (GCN) networks and in particular graph-to-graph neural networks (GGN). The final aim will be to attribute atoms' influence on properties through a perturbation based analysis.

The following sections will give an introduction about the background such as Med Chem and molecules representation; then an analysis machine learning applications to drug design and GCN will follow and finally the proposed method will be presented together with results and comparisons with already existing XAI methods.

# Chapter 2

# Related Work

This section introduces the field of drug design and the main related computational approaches. In particular, importance will be given to the representation of molecules and the following applications.

Successively, an overview on Explainable AI will follow and further details on state of the art methods will be analysed.

## 2.1 Drug Discovery

Drug Discovery is a process consisting in searching, developing and optimizing drugs in order to disclose new lead compounds for future new medicaments for specific diseases.

Once chosen a particular disease, drug discovery process typically starts with the identification and validation of a specific macro-molecular target, which generally consists in a protein. Subsequently, screening and optimization procedures follow, in order to identify hits and leads that satisfy a set of predefined criteria.

Finally, the lead compound development involves a multi-factorial optimization of potency, selectivity (pharmacodynamics) and pharmacokinetics properties such as absorption, distribution, metabolism, excretion and toxicity, the latters usually referred as ADMET.

One big issue in drug discovery consists in experimentally evaluating the molecules, a part of the drug development flow that is time-consuming and expensive. In order to compensate this problem, medicinal chemists generally use computational approaches to derive models for forecasting molecules' bio-activity and selecting the ones associated to higher predicted biological potency.

Two main computational drug design approaches are:

- Structure based drug design (SBDD): both ligands and proteins are taken into account trying to study the mechanism of activation or inhibition of the target. At Med Chem Level this approach generally involves techniques such as molecular docking and comparative molecular binding energy analysis (ComBinE) which try to find out where the ligand binds to the target and which parts of both actors interact with either positive or negative contribution.
- Ligand based drug design (LBDD): only small molecules (i.e. the drugs) are considered to individuate particular structural patterns which likely determine the drugs' potency.

LBDD is the approach this study will focus on, developing statistical models to predict and explain the activity of molecules in order to create and select new compounds to acquire/synthesize and evaluate biologically.

## 2.2 AI for Drug Design

AI has begun to be one of the most powerful tool for drug discovery nowadays. It is not only used for predicting molecules' properties and activity, but some recent studies exploited NN also for generating new molecules maximizing or minimizing a given scoring function (Zhavoronkov et al. [2019], Popova et al. [2019], Popova et al. [2017]).

### 2.2.1 Molecules Representation

One of the most important aspect of using AI in Med Chem is determining a way to represent the under study molecules. Classical approaches include two main methods for solving this problems:

- Chemically: chemical and physical/chemical molecular descriptors can be calculated to describe molecules' properties (ex. molecular weight, Number of bonds). It is in fact known since the sixties (Hansch et al. [1962]) that these properties are generally highly correlated with the drugs' potency;
- Structurally: a different way to describe the molecules implies the use of molecular fingerprints, that consist in the generation of bit-vectors in which each element indicates the presence or the absence of one atom or group. One popular method of the latter representation are Morgan Fingerprints or Extended Connectivity Fingerprints (ECFPs) (Rogers and Hahn [2010]). This method can also be connected with that of Free and Wilson [1964], a couple of medicinal chemists who were the first to use indicator variables as a sort of fingerprints.

### Morgan Fingerprints and Graphs

Morgan Fingerprints, also referred as Circular Fingerprints, consist in a particular algorithm for generating molecular fingerprints. The procedure for generating fingerprints as following:

- **Assign each atom an identifier**: For each atom in the molecule some properties (invariants) such as the atomic number, the mass, the number of hydrogens and the number of bonds are calculated. Then all the values are hashed into an integer, this number will be the identifier for the atom. It is easy to get that same atoms will have the same identifier, but, for instance a carbon connected to two hydrogens will have a different identifier than an another carbon connected only to one hydrogen. The calculated identifiers are then stored in a set.
- **Update each atoms' identifiers based on its neighbours**: For each atom a list of the identifiers of its neighbours is stored, together with an integer identifying the bond type. Once the list is obtained, it is then hashed again into an integer. This value will be the identifier for the atom at the second iteration. This process is repeated $r$ times, where $r$ is the radius, that is normally related to the number of connecting bonds from the starting atom ($r = 1$ means one bond radius, $r = 2$ two bonds and so on). In fact, using $r = 2$ implies identifying all the molecule substructures of radius up to 2. Generally the most common value for $r$ would be 2.
- **Store each identifier in an array**: All the obtained identifiers throughout the different iterations have to be stored in a bit-vector, which is generally of size 2048. Obviously for doing this operation it is needed to take advantage of the modulo operator.

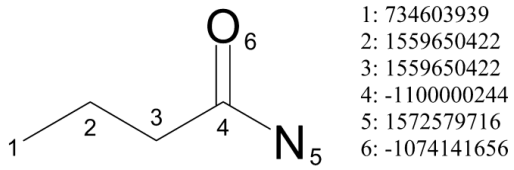



**Figure 2.1.** Atom Identifiers through iterations applied to the butyramide. Image reproduced as published by Rogers and Hahn [2010]. Hydrogen atoms are not shown as they are usually embedded into the connected heavy atoms (non hydrogens).

As an example the molecule of butyramide is considered as shown in Figure 2.1. Each atom is associated to a number and the identifiers are calculated hashing the invariants as explained above. The procedure then follows, for the atom 4, a list is

initialized as [(1, -1100000244)] (the first integer, and, for each non-hydrogen neighbor we append to the list a couple (i, j), where i is the bond identifier, 1,2,3,4 for single, double, triple and aromatic bonds, and j is the neighbor identifier). The resulting list for atom 4 is the following: [(1, -1100000244),(1,1559650422),(1,1572579716),(2,-1074141656)]]. Atom 4's identifier is then produced hashing the obtained list: -1708545601.

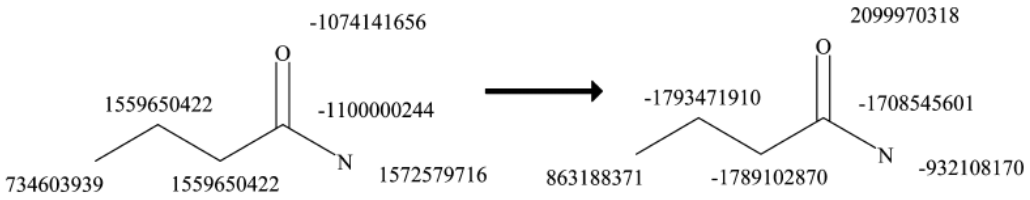



**Figure 2.2.** Atom Identifiers through iterations. Image reproduced as published by Rogers and Hahn [2010]

This idea brings to a way of representing molecules using graphs. Graphs turn out to be very effective for molecules since they allow to represent together chemical (such as atom's mass, number of bonds, hybridization) and structure information of the drugs.

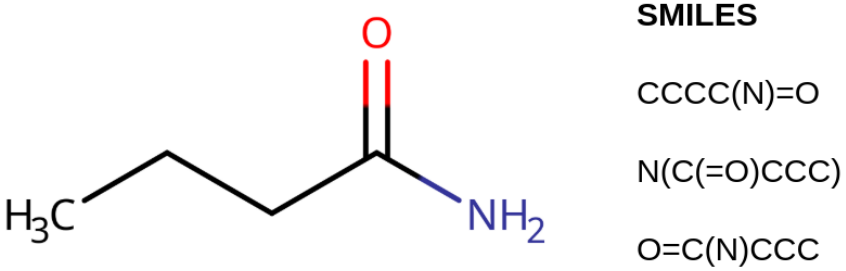



**Figure 2.3.** Chemical structure of butyramide with explicit hydrogens on terminal atoms and depiction of the corresponding SMILES. Three SMILES notations can be written for the butyramide

Another important way of representing molecules is the simplified molecular input line entry system (SMILES), which is a sequential textual representation of atoms' connectivity of molecules. One problem with this representation is its non-uniqueness, a molecule could, in fact, be represented in multiple ways. Figure 2.3 shows the

butyramide structure and three different SMILES string that represent it. Being the SMILES a textual representation, it is possible to exploit this encoding through recurrent neural networks (RNN), which are known to be powerful when dealing with sequential inputs. The ambiguity and the lack of chemical information in the SMILES representation are a significant drawback for the usage of RNN. Although it is not trivial to use SMILES for drug design, there are some studies who managed to obtain good results in different tasks such as molecule generation (Popova et al. [2017]).

The approached proposed by this study will make use of the graph representation of molecules. This, as already said, allows to exploit both the structure of the molecule and atomic chemical features such as the weight, the charge and the number of bonds. Moreover the graphs do not suffer from the ambiguity problem described for the SMILES.

Given a molecule with $n$ atoms connected by $m$ bonds it is always possible to translate it into a graph with $n$ atoms connected by $m$ edges. In particular, a graph is represented by two main matrices:

- The adjacency matrix $A_{n\times n}$, in which each element $\{i, j\}$ is 1 if nodes $i$ and $j$ are connected else 0.
- The features matrix $X_{n\times m}$, which consists in the features of each node.

Using the Open Source Software RDKit (Landrum) it is possible to gather 30 atom features summarized in Table 2.1

| Feature | Type |
|---|---|
| Atomic Number | Integer |
| Formal Charge | Float |
| Degree | Integer |
| Depleted Degree | Integer |
| Full Degree | Integer |
| Valence Degree | Integer |
| # Hydrogens | Integer |
| # Implicit Hydrogens | Integer |
| # Instanced Hydrogens | Integer |
| # Total Hydrogens | Integer |
| # Valence Electrons | Integer |
| Explicit Valence | Float |
| Implicit Valence | Float |
| Valence | Float |
| Chirality | One Hot Encoded Vector |
| Is Terminal | Boolean |
| Is Aromatic | Boolean |
| Is in Ring | Boolean |
| VdW Radius | Float |
| VdW Volume | Float |
| Covalent Radius | Float |
| Ionisation Energy | Float |
| Electron Affinity | Float |
| Principal Quantum Number | Float |
| Polarisability | Float |
| Pauling Electronegativity | Float |
| Sanderson Electronegativity | Float |
| Kier Hall Electronegativity | Float |
| Mc Gowan Parameter | Float |
| Kier Hall Alpha Contrib | Float |

**Table 2.1.** Atom Features Table.

### 2.2.2 QSAR

QSAR (or QSPR) consists in the application of statistical and machine learning models for predicting molecules activities (or properties) from their structures. Molecules' activity is generally represented in terms of drug concentration needed to inhibit or activate a certain target. A molecule is considered "more potent" if it is needed lower amount (concentration) in order to bind the molecular target.

Since most of the times the concentration is expressed using powers of ten, it is often converted using the -log, in a similar fashion as the well known pH to evaluate the acidity of a solution. One of the most used activity representation is the $IC_{50}$, which is the concentration needed to inhibit the 50% of the target. Using the -log transformation the so $pIC_{50}$ is calculated. Using the -log operator, not only linearizes the scale of the activities, but it also inverts it, having higher $pIC_{50}$ with more potent drugs.

One big flaw of QSAR is the fact that bioacitivity data often comes from different sources and laboratories, this unfortunately generates a lot of noise and outliers.

The most used algorithms for generating models are Random Forest, SVM and Linear Regression. As already mentioned the development of Recurrent Neural Networks and Graph Neural Networks have increased the usage of more complex models to predict the molecules activity.

Generally the most used evaluation method for QSAR models are the determination coefficient $r^2$ and the standard error of prediction (SDEP).

#### 3-D QSAR

QSAR, also known as 2-D QSAR does not take into account any spatial information of the molecules. Inclusion of molecular conformational data, leads to the definition of 3-D QSAR. This technique consists in calculating molecular interaction fields (MIFs) between the molecule and a probe atom in the space and uses these as features for building the models. In particular, two main types of MIFs are considered:

- Electrostatic field: The Coulomb force between all the atoms of the molecule and the probe atom is calculated.
- Steric field: The Lennard-Jones potential between all the atoms of the molecule and the probe atom is calculated.

The choice of the probe atom consists in an additional parameter of the model.

It is important to notice that in order to generate consistent interactions among the dataset, all the molecules need to have the common substructures aligned, in order to be able to find correlation between substructures and property changes.

The usage of MIFs with Partial Least Squares (PLS) modelings is known as Comparative Molecular Field Analysis (CoMFA), the first 3-D QSAR (Cramer et al. [1988]). Multiplying the PLS coefficients with the interactions calculated on each molecule it is possible to study the most important interactions, generating the so called Activity Contribution plots. This operation is somehow a precursor of XAI, considering it was performed for the first time in the '80s.

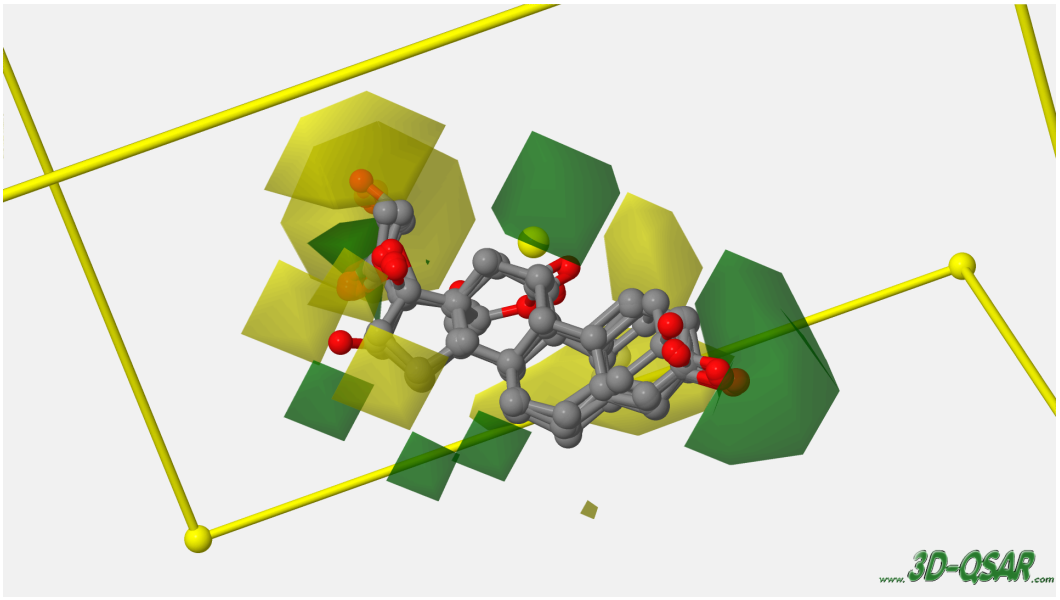



**Figure 2.4.** Visualization of PLS coeffs times mean steric interactions

Figure 2.4 shows the aligned molecules of a dataset together with the plots of the PLS coefficients multiplied by the mean values of the interactions of all the molecules. This plot gives a general idea of the structure of the whole dataset showing the substructures which generally contribute positively (green) and negatively to the activity using steric interactions. Figure 2.6, instead, shows the activity contribution plots for one molecule using electrostatic interactions: this plot allows to study for each single molecule the parts that contribute positively (blue) and negatively (red) to the activity, in this case, the analysis can also give some information on how to modify hydrogen bonding abilities (donor or acceptor) around the molecules.

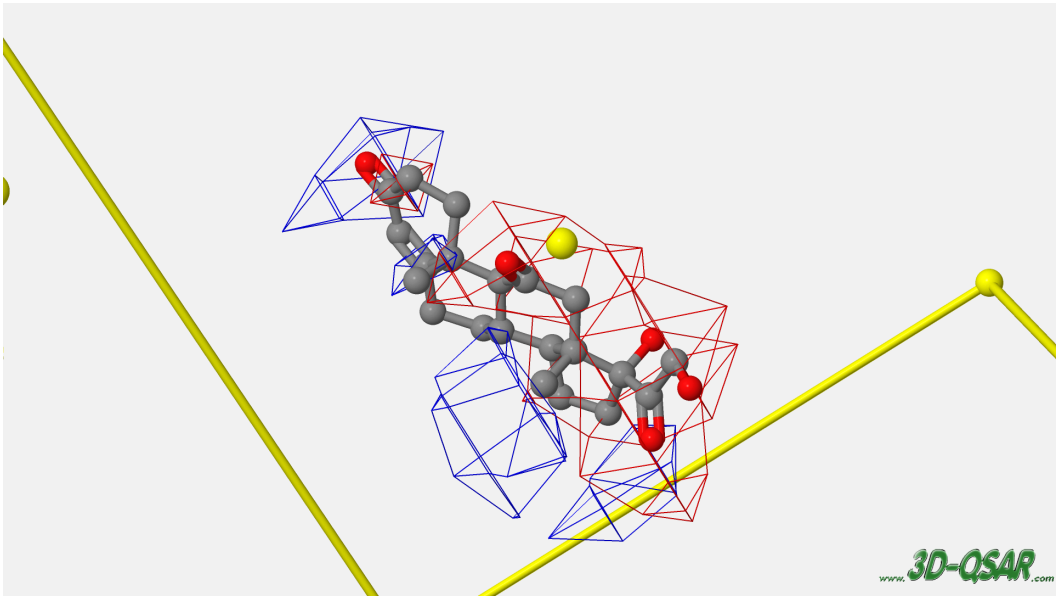



**Figure 2.5.** Visualization of PLS coeffs times electrostatic interactions

As already mentioned, 3-D QSAR, and in particular CoMFA, represents one of the first and most efficient ways for visualizing Structure-Activity relationships. Unfortunately, there are some limitations which make 3-D QSAR not very welcomed by a part of the Med Chem community:

- The models highly depend on how the dataset molecules are aligned.
- In its original implementation CoMFA uses only one conformation per molecule. Using different conformations might affect the alignment procedure and 3-D QSAR model performances, this is somehow implemented in 4-D QSAR methods.
- Considering the MIFs calculation, the application to a large dataset is somehow limited as a big amount of volatile memory is required to avoid data swapping a thus long calculation time. The usage of GPUs has softened this limit but having large datasets not only affects calculations of MIFs but also of the alignment procedure.

The plots in Figure 2.4 and 2.6 were generated using models generated on the `www.3d-qsar.com` platform (Ragno [2019]) using the original dataset of the first CoMFA paper by Cramer et al. [1988].

## 2.3 Graph Convolutional Neural Networks

Graph Convolutional Neural Networks (GCN) (Kipf and Welling [2016]) are powerful techniques which enable to learn over graphs representations making nodes features flow through edges of a graph structure. The concept of convolution comes from computer vision. The convolution operator consists of replacing the pixel of an image with weighted sum of its neighbors. Taking inspiration from the convolution concept, in graph convolution, the values of one node are replaced with a weighted sum of its neighbors.

In particular, given a graph represented using an adjacency matrix $A$ and a features matrix $X$, the vanilla GCN layer performs the following calculation:

$$y = \tilde{A}XW + b$$

where $W$ is the matrix of the learnable weights, $b$ are the biases and $\tilde{A} = A + I$ is the adjacency matrix in which also self-loop edges for all the nodes are taken into account.

A more sofisticated technique is to normalize the adjacency matrix using the degree of each nodes (i.e. the number of connections). In particular the equation above is rewrited as:

$$y = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}XW + b$$

where $\tilde{D}$ is the degree matrix of the adjacency matrix $\tilde{A}$.

Using multiple sequential GCN layers allows to spread information in the graph at different radius similarly as it happens with Morgan Fingerprints.

Depending on the output, GCN networks can perform different tasks. In this study only two of them will be investigated:

- Graph classification/regression: if the output is only one value (or more, for multiple classes/values), it is possible can perform classification and regression on the graph. In this task, generally, a latent vector representation of the graph from the nodes is generated using the maximum and average operators among all node features. This approach will be used in this study for predicting molecules properties and perform a variant of QSAR using graphs (GCN-QSAR).
- Node classification/regression: Instead of grouping the node features into one single vector, it is possible to have an output for each node. This task is generally performed in social networks contexts for assigning nodes to particular groups. One common approach for this task is the semi-supervised learning, which consists in masking some “test nodes” during the training phase for predicting them later during testing. This particular method will be a key

concept behind the proposed approach for attributing molecules activities to atoms.

### 2.3.1 Global Pooling Layers

GCN layers work on node-wise features propagating them through the edges. When dealing with graph classification/regression, it is necessary to encode the node features into graph features. Once the graph features are calculated, they can be fed through linear layer which decrease the shape to the output size.

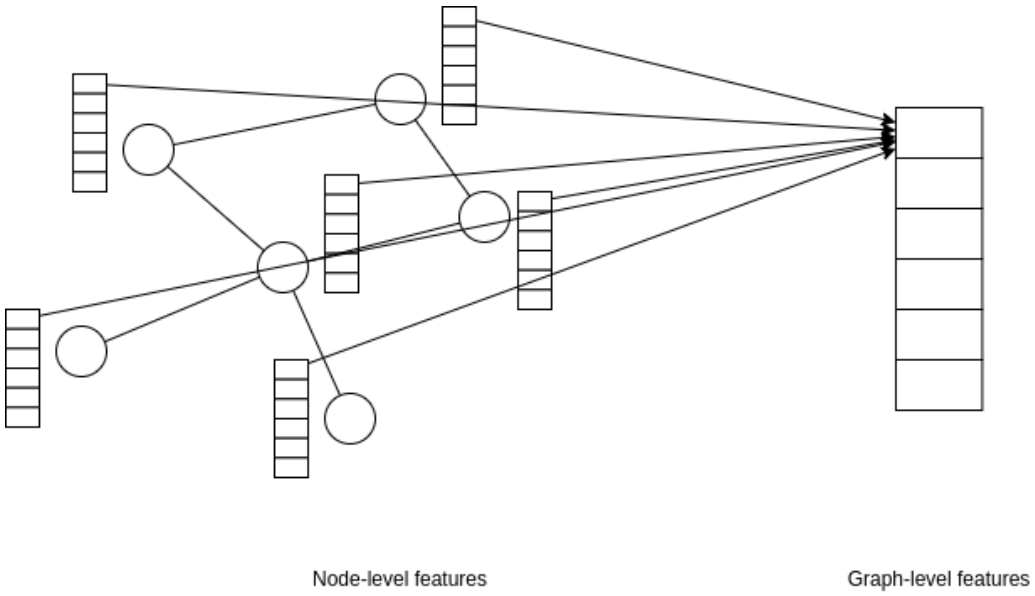



**Figure 2.6.** Graph pooling layer sketch

The graph vectorization is generally performed using pooling layers, in fact, while with images it is possible to reshape the image matrix into one vector, with graphs of varying sizes, it is different. What generally happens is that the vector is formed using the maximum or average values for each feature of the nodes. These techniques are generally referred as global max pooling (GMP) and global average pooling (GAP).

### 2.3.2 PyTorch Geometric

PyTorch Geometric (Fey and Lenssen [2019]) is one of the most used extension library for geometric deep learning in PyTorch. It consists in several methods for

deep learning on graphs and other irregular data types.

PyTorch Geometric handles graphs and other structures through the object Data. Data contains three main attributes:

- node features: an array of shape [num_nodes, num_node_features] which contains all the features of the nodes
- nodes connections: described using an array of shape [2, num_edges] in which the indexes of the nodes of the directed edge are specified
- target value: this value could have different shapes depending on the task. For instance in case of node classification the resulting shape would be [num_nodes, 1]

```
import torch
from torch_geometric.data import Data

edge_index = torch.tensor([[0, 1, 1, 2],
                           [1, 0, 2, 1]], dtype=torch.long)
x = torch.tensor([[-1], [0], [1]], dtype=torch.float)

data = Data(x=x, edge_index=edge_index)
>>> Data(edge_index=[2, 4], x=[3, 1])
```

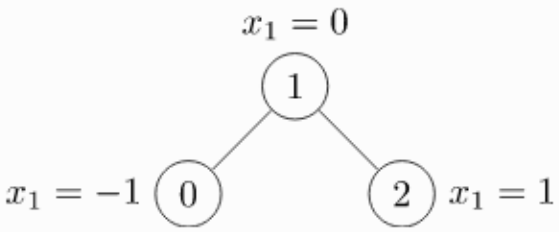



**Figure 2.7.** PyTorch Geometric Example

Figure 2.7 shows an example code which creates a graph of three nodes with two edges using PyTorch Geometric.

Since in deep learning training is generally performed in batches, PyTorch Geometric implements the Batch object which stores all the graphs into a single object. This allows having multiple disconnected graphs encoded into one and makes calculations much faster.

All the experiments later provided in this thesis will make use of the PyTorch Geometric implementation of GCNs and graphs operations.

## 2.4 Explainable AI

Neural networks (NNs) implementation in AI improved its power such that computers are starting to make choices for our own. The ability of AI to make decisions leads to the fact that these choices need to be explained or interpreted. This lack is also boosted by the inclination of NN nature to be Black Box architectures. In fact, while classic machine learning models such as decision trees and linear models have few parameters that can be interpreted easily, NNs imply the usage of thousands of layered parameters, implicitly difficult to explain.

XAI aims to develop readable models and create some techniques to generate interpretable models while maintaining high performance levels. The term was first coined by Lent et al. [2004] to show the ability of their system to explain the choices of AI agents in games and army simulators.

### 2.4.1 XAI approaches

There are different ways to approach the problem of generating human interpretable explanations of the models' decisions. XAI techniques can be classified in two dimensions (Rai [2020]): (i) whether the technique is model-specific or model-agnostic and (ii) whether the technique is global or local.

The former category separates the methods which are only applicable to some specific algorithms from general methods, while the latter classifies methods which are able to give an explanation with all the inputs or with specific ranges of inputs.

#### Model-specific global explanation

These methods try to increase the interpretability of the models imposing structural constraints like sparsity and monotonicity. An example of this class of XAI is the concept of Restraining Bolt (Giacomo et al. [2019]), which is a reinforcement learning technique that constraints the agent using temporal logic specifications using the LTLf and LDLf language. This is important as guiding the agent while learning through logical specifications, not only allows to learn hard sequential tasks, but gives also the chance to avoid learning weird patterns in the data that could instead be found using only random exploration.

#### Model-specific local explanation

These techniques try to explain specific instances for deep learning models. For instance, attention based mechanisms belong to this class of XAI.

### Model-agnostic global explanation

This class of XAI methods try to generate an explainable "glass box" model that approximates the black box network. For example an interpretable decision tree model could be generated starting from the initial model.

### Model-agnostic local explanation

This category aims to generate an explanation of the output of a model locally for certain inputs. An important technique is LIME (Ribeiro et al. [2016]), local interpretable model-agnostic explanation, which generates an explanation of a model's output in the "neighborhood of an instance".

LIME generates an explanation by approximating the model locally with an interpretable one. Interpretable models are linear models with strong regularization, decision trees etc, trained on small perturbations around the original portion of data.

This is an important drawback, since linear models might be able to only approximate the original model only on a small portion of data. Using the same weights for very different data might lead to wrong results.

### 2.4.2 Salience Mapping

A common approach is to understand the representation of a deep network and address which part of the input is important to the generation of the output using salience mapping. In particular, the aim is to generate an attribution of a prediction at a certain input with respect to a baseline, which is, an object with the same shape of the input in which each element in position $i$ is the contribution of the input $x_i$ to the output with respect to the baseline.

Common techniques of this approach are Gradients, LRP (Bach et al. [2015]), DeepLIFT (Shrikumar et al. [2017]), CAM (Zhou et al. [2015]), Grad-GAM (Selvaraju et al. [2019]) and Integrated Gradients (Sundararajan et al. [2017]).

#### CAM and Grad-CAM


**Figure 2.8.** CAM Explanations

Class Activation Mapping (CAM) is an approach which adds a global average pooling layer followed by a softmax in convolutional neural networks. While Grad-CAM (gradient-weighted class activation mapping) is an improvement that generalizes CAM using the gradients of the target to highlight important regions in images. Figure 2.8 shows some explanations obtained by CAM developers. It is possible to see how the model is able to classify the image with "brushing teeth" labels and at the same time the most relevant part of the image is the tooth brusher.

An important drawback of these methods is that in order to work they edit the architecture of the model leading to lower performances.

#### LRP

Layer-wise relevance propagation (LRP, Bach et al. [2015]) is a particular method for generating saliency maps which consists in propagating backward through the

network "relevance values" $R$ for each layer.

In particular the basic LRP rule for propagating the relevance between two neurons $j$ and $k$ of any consecutive layers is:

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k \tag{2.1}$$

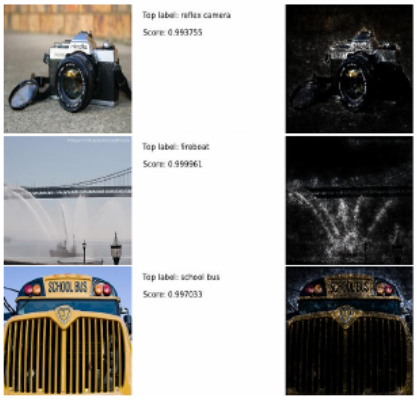
### DeepLIFT

Deep learning important features (DeepLIFT, Shrikumar et al. [2017]) is the first method to introduce the concept of reference input. In fact, with this method attributions are generated with respect to the input compared to a baseline.

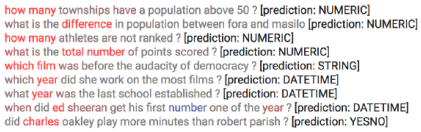
The main concept behind DeepLIFT is the propagation of attributions in a backward fashion from the output to the input. In particular, attributions are scaled by the difference between the output and the reference output.

One important aspect that has to be taken into account is the choice of the baseline. For pixel features it is generally common to use a black image while for vector features it is common to use zero vectors. In this study, the baseline for molecules is a graph equal to the molecule but with zero vector features.

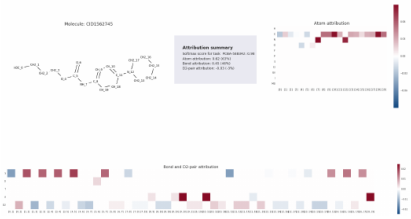
### Axiomatic Approach




(a) Image Explanation




(b) Text Explanation



(c) Molecule Explanation

**Figure 2.9.** Explanations of Integrated Gradients

Sundararajan et al. [2017] proposed an axiomatic approach to this solution developing a method called Integrated Gradients (IG). This solution is based on integrating the gradients of the output with respect to the input on a path from the baseline to the input itself. It is important to notice that this does not need to add any additional structure to the original model changing its performance.

Also IG uses the concept of baseline input for the calculation of saliency maps, so as for DeepLIFT it is important to choose the right reference input for the type of data.

In particular Sundararajan et al. [2017] claim that most of the common used methods such as DeepLIFT and deconvolutional networks break two important axioms that each method should satisfy:

- Sensitivity: an attribution method satisfies Sensitivity if for every input and baseline that differ in one feature but have different predictions then the differing feature should be given a non-zero attribution. A common method like the gradients one breaks sensitivity since in presence of ReLU, the input

might be flattened and receive zero gradient.

- Implementation invariance: if two models are functionally equivalent (i.e. to equal inputs return equal outputs despite having different implementation), they should give same attributions to the inputs.

  In this case, the gradients method satisfies implementation invariance thanks to the derivative chain rule, but approaches like DeepLIFT and LRP break implementation invariance because they use a form of discrete gradients, in which, in general, the chain rule does not hold.

IG method satisfies both sensitivity and implementation invariance and it is defined as

$$IG_i(x) = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha) \times (x - x')}{\partial x_i} d\alpha$$

.

Since IG method is based on path integral, it satisfies implementation invariance. This method is proved to satisfy one more axiom called completeness, which is, the property such that the attributions add up to the difference between the output of F at the input $x$ and the baseline $x'$. Since completeness implies sensitivity, also the latter is satisfied.

### 2.4.3 Captum

Captum (Kokhlikyan et al. [2020]) is an open-source extension library for PyTorch which implements a variety of XAI methods.

Captum provides state-of-the-art algorithms, including IG, to provide researchers and developers a systematic way for generating explanations of models.

For model developers, Captum can be used to improve and troubleshoot models by facilitating the identification of different features that contribute to the output.

The main feature of Captum it's its flexibility which allows developers to use any kind of input representation with the several provided algorithms.

```
>>> # ImageClassifier takes a single input tensor of images Nx3x32x32,
>>> # and returns an Nx10 tensor of class probabilities.
>>> net = ImageClassifier()
>>> ig = IntegratedGradients(net)
>>> input = torch.randn(2, 3, 32, 32, requires_grad=True)
>>> # Computes integrated gradients for class 3.
>>> attribution = ig.attribute(input, target=3)
```

**Figure 2.10.** Captum example code

In Figure 2.10 it is shown an example code for calculating Integrated Gradients attributions on image predictions of a classifier. Captum allows to calculate explanations with just 4 rows of code and it is not only suitable for image data types but also for text, graphs etc.

The experiments performed in the next sections will make use of Captum for providing comparison with the proposed method for attributing molecules activity.

# Chapter 3

# Proposed Approach

This chapter defines the proposed method for attributing atoms for explaining structure-activity relationship learned through a previously trained QSAR model. This study makes use of the graph representation of the molecule and, in particular, of a semi-supervised approach. The idea behind this method is to give positive values to atoms which contribute positively to the activity and negative values to atoms which contribute negatively to the activity.

Given a molecule $m$ and a second molecule $m'$, which is obtained adding or substituting one or more atoms to $m$, the set of atoms added or substituted from $m$ to generate $m'$ is defined as $g = m' - m$ , while $a$ is the activity of the molecule $m$ and $a'$ is the activity of the molecule $m'$. The attributions of the atoms $g_i \in g$ will be defined as:

$$attr(g_i) = \frac{a' - a}{|g|} \tag{3.1}$$

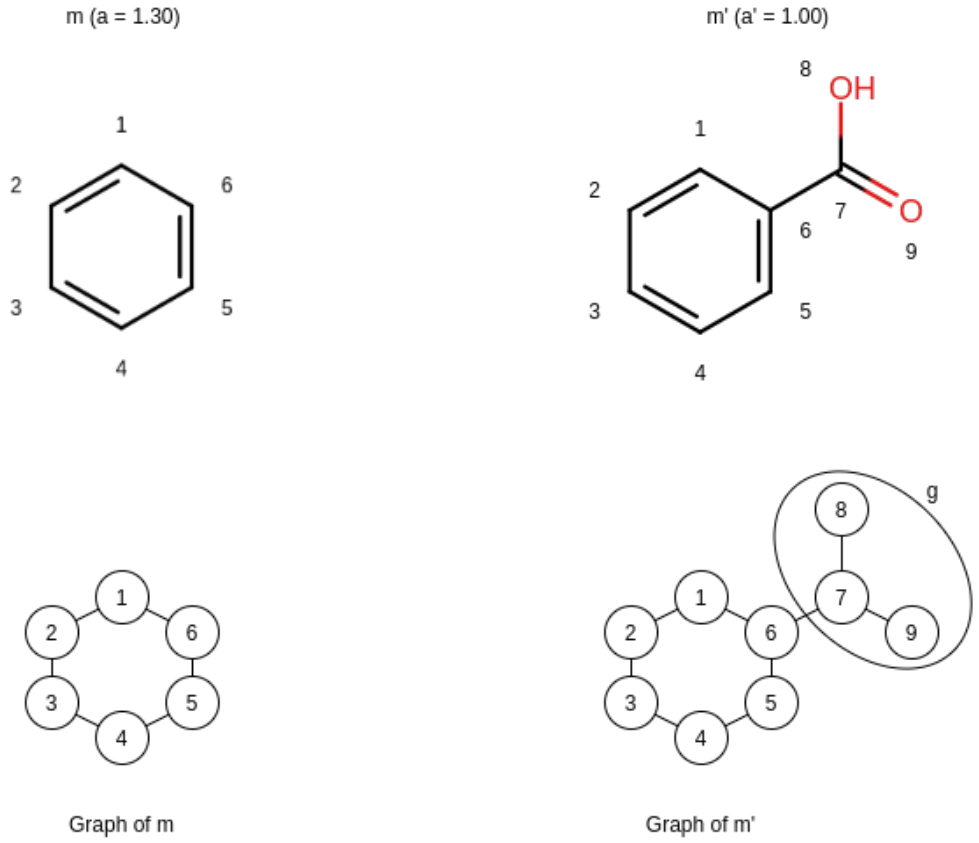



**Figure 3.1.** Molecule generation: the benzoic acid is generated adding the carboxyl group to the benzene

Figure 3.1 gives an example of the procedure just explained. Assuming, as in Figure 3.1, that the activity of $m$ (benzene) is $a = 1.30$ and the activity of $m'$ (benzoic acid) is $a' = 1.00$, the attribution model should be able to assign to each atom of $g$ (carboxyl group) (in this case $g_i \in \{7, 8, 9\}$) the value $attr(g_i) = \frac{1.00-1.30}{3} = -0.10$.

Intuitively this method attributes positively the atoms of $g$ if they increased the activity of the molecule, otherwise negatively.

Given a dataset $D$ formed by $N$ molecule-activity couples $(m, a)$, the workflow is the following:

- Train QSAR (or GCN-QSAR) Model on $D$.
- For each molecule $m$ in the dataset generate $M$ molecules $m'_i$ by adding or replacing one group and form a new dataset. Store all the $n$ couples $(m, m'_i)$ in a dataset.
- For each couple $(m, m')$ in the dataset use the QSAR model to predict $\hat{a}$ and

$\hat{a}'$, calculate $g = m' - m$ and $\hat{a}' - \hat{a}$ and finally store $attr(g_i)$ in the dataset.

- For each triple $(m, m', attr(g_i))$ train a GGN that takes as input the graph of $m'$ and produces as output a graph in which the nodes corresponding to $g$ have value $attr(g_i)$.

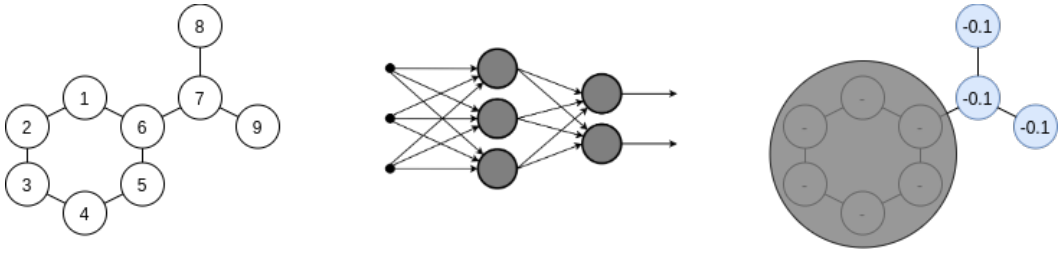



**Figure 3.2.** XAI Model Training

Figure 3.2 is a sketch of the training phase for the example of the benzene and benzoic acid. The gray overlay represents the fact that during training, for each molecule training is done only on the atoms of $g$. This task is performed calculating the Mean Squared Error (MSE) loss only on the atoms of $g$.

It is important to note that not training on the output of the other atoms, still takes into account them since thanks to the concept of graph convolution. In fact, the value of the atoms of $g$ depends also on the features of their neighbors.

Figure 3.3 shows the part of code to train only on the added/substituted atoms of the molecule using a bit-vector mask while calculating the loss.

```python
for m1,m2,a in dataset:
    # Get number of atoms in m2
    atoms = m2.GetNumAtoms()

    # Get common atoms between m1 and m2
    matching_atoms = m2.GetSubstructMatch(m1)
    added_atoms = [a for a in range(atoms) if a not in matching_atoms]

    # Generate the target output
    target = torch.zeros(batch_atoms).float()
    target[added_atoms] = a

    # Generate graph using torch geometric
    graph = Batch.from_data_list([gen_graph(m2)])
    x, edge_index, b = graph.x, graph.edge_index, graph.batch

    # Generate explainations using the model
    explain = model(x.to(device), edge_index.to(device), b.to(device)).flatten()

    # Mask to calculate the loss only on added/substituted atoms
    mask = target != 0

    # Calculate the loss and backpropagate
    l = loss(explain[mask], target[mask])
    optimizer.zero_grad()
    l.backward()
    optimizer.step()
```

**Figure 3.3.** XAI Model Training Code

## 3.1 Molecule Generation

In order to generate new molecules a set of substituents to attach or exchange to the atoms of the molecules is needed.

Ertl [2017] proposed an algorithm to extract functional groups from a set of molecule and applied it to $\approx$ 483,000 molecules with activity below 10 $\mu$M on any target retrieved from ChEMBL (Gaulton et al. [2012]) resulting in a set of 768 FGs that are present in at least 10 ChEMBL molecules is provided in pseudo-SMILES notation.

### 3.1.1 Functional Groups Extraction Algorithm

It is interesting to give a look at the algorithm for extracting the functional groups, which can be summarized in the following steps:

- mark all heteroatoms in a molecule, including halogens
- mark also the following carbon atoms:
  - atoms connected by non-aromatic double or triple bond to any heteroatom
  - atoms in nonaromatic carbon–carbon double or triple bonds
  - acetal carbons, i.e. sp3 carbons connected to two or more oxygens, nitrogens or sulfurs; these O, N or S atoms must have only single bonds
  - all atoms in oxirane, aziridine and thiirane rings (such rings are traditionally considered to be functional groups due to their high reactivity).
- merge all connected marked atoms to a single FG
- extract FGs also with connected unmarked carbon atoms, these carbon atoms are not part of the FG itself, but form its environment.

The algorithm described above iterates only through non-aromatic atoms. Aromatic heteroatoms are collected as single atoms, not as part of a larger system. They are extended to a larger FG only when there is an aliphatic functionality connected (for example an acyl group connected to a pyrrole nitrogen). Heteroatoms in heterocycles are traditionally not considered to be "classical" FGs by themselves but simply to be part of the whole heterocyclic ring.

### 3.1.2 Randomized Generation

Once the functional groups database (FGDB) is generated, it is possible to perturb the molecules using a breadth-first algorithm as illustrated in Figure 3.5. In particular, given two parameters N and D, starting from the first molecule, N random moves are selected (a move consists in attaching or replacing a group to a random atom of

the molecule); this procedure is then performed D times on the generated molecules at the step before. In this way, the original molecule is not only perturbed once, but also multiple times.

For instance, Figure 3.6 shows an example of generation using N=2 and D=2: starting from the 4-ethylaniline the algorithm randomly selects 2 functional groups from the FGDB and inserts them in two random positions, generating two new molecules. The operation is the repeated on the generated molecules in order to form 4 more molecules.

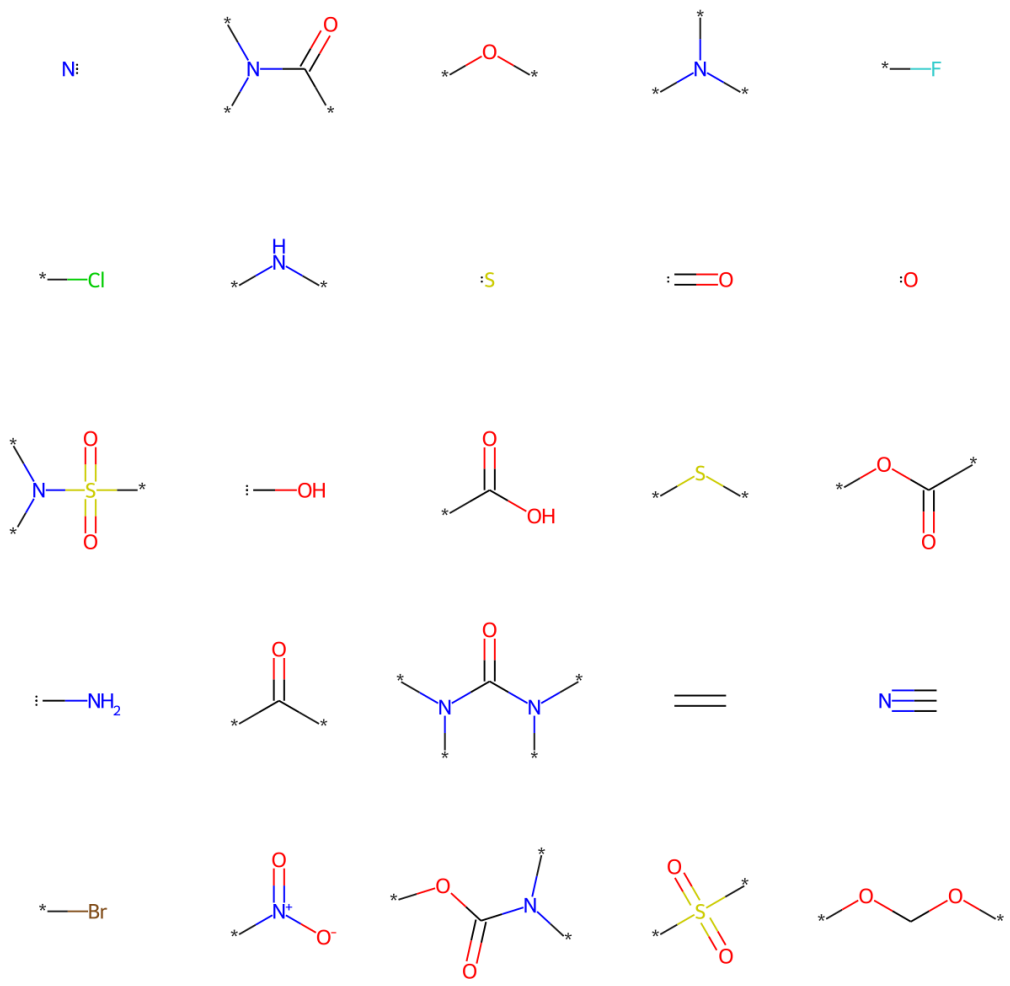



**Figure 3.4.** 25 functional groups from Ertl [2017] Database

```python
def generate_moves(self, initial_mol, N, D):
    generated = []
    to_visit = [(initial_mol, 0)]
    while len(to_visit)>0:
        mol, l = to_visit.pop(0)
        moves = []
        self.candidate_atoms = self.get_candidate_atoms(mol)
        for i in range(N):
            move = None
            attempts = 0
            while move is None or move in moves:
                if attempts > N+10:
                    move = None
                    break
                try:
                    move = self.get_random_move()
                    new_mol = self.apply_move(mol, move)
                    smiles = Chem.MolToSmiles(new_mol)
                    new_mol = Chem.MolFromSmiles(smiles)
                except:
                    new_mol = None
                if new_mol is None:
                    move = None
                attempts += 1
            if move is not None:
                moves.append(move)
                generated.append((mol, new_mol))
                if l<D:
                    to_visit.append((new_mol, l+1))
    return generated
```

**Figure 3.5.** Molecule Perturbation Algorithm

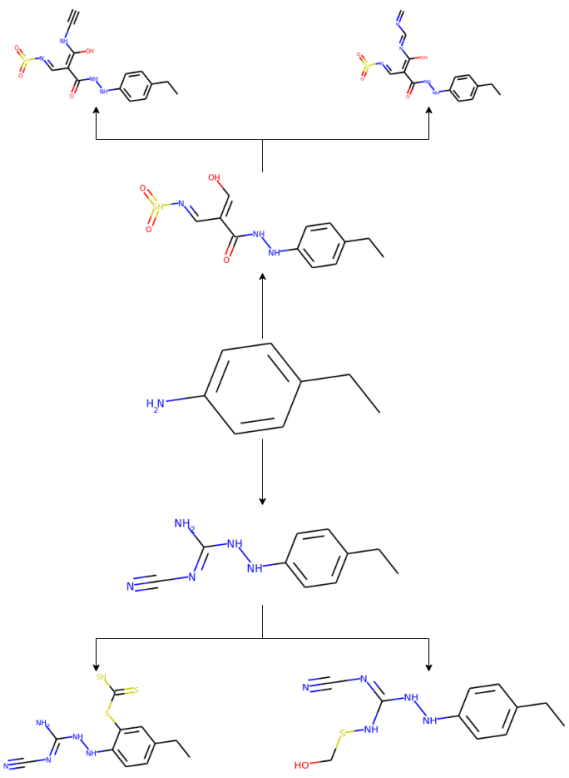


**Figure 3.6.** Example of generation procedure using N=2 and D=2 starting from the 4-ethylaniline

## 3.2 Experiments

This section analyses the several experiments performed in order to test the proposed method for explaining Structure-Activity relationship.

The method has been validated in two main problems:

- Water Solubility Prediction
- pKa Prediction

The problems work as a benchmark since, in order to evaluate the results, the experiments check the coherence of the attributions with the chemistry theory.

### 3.2.1 Water Solubility Prediction

According to the IUPAC definition, solubility is the analytical composition of a saturated solution expressed as a proportion of a designated solute in a designated solvent. Solubility may be stated in various units of concentration such as molarity, molality, mole fraction, mole ratio, mass (solute) per volume (solvent) and other units.

Water solubility is very important in Med Chem since knowing whether a molecule can be solubilized in water or not is crucial while developing drugs or substances which are designed to be administered orally. Solubility constitutes a major challenge in pharmaceutical chemistry and drug optimization. It is estimated that about 40% of new chemical entities developed are practically insoluble. Since drugs must be present in the form of solution at the site of absorption, it is often crucial to enhance molecules solubility. This process, which generally involves a lot of trials and errors, increases the cost of developing drugs and very often solubility is also related to ADMET profiling. A bad ADMET profile is the major cause of failure in the last stages of clinical trials, the most expensive part of a drug developing process. Studying the relationship between structure and solubility through XAI could help containing these expenses and lowering clinical trial failures.

A popular aphorism used for predicting solubility is "like dissolves like" also expressed in the Latin language as "Similia similibus solvuntur". This statement indicates that a solute will dissolve best in a solvent that has a similar chemical structure to itself. This view is simplistic, but it is a useful rule of thumb. The overall solvation capacity of a solvent depends primarily on its polarity. For example, a very polar (hydrophilic) solute such as urea is very soluble in highly polar water, less soluble in fairly polar methanol, and practically insoluble in non-polar solvents such as benzene. In contrast, a non-polar or lipophilic solute such as naphthalene is insoluble in water, fairly soluble in methanol, and highly soluble in non-polar benzene.

The concept of polarity helps in predicting the water solubility of the molecules and attributing it to the atoms. In fact, highly electronegative atoms such as Oxygen, generally increase solubility since they increase polarity.

In the following tests, it is expected to have the proposed model to attribute positively atoms such as oxygen and negatively chains of carbons which are generally non-polar.

### Dataset

ESOL (Delaney [2004]) is a standard regression dataset containing structures and water solubility data for 1128 compounds. The dataset is widely used to validate machine learning models on estimating solubility directly from molecular structures (as encoded in SMILES strings).

The preprocessed dataset was downloaded from Kaggle (sol) and was used for generating a GCN-QSAR model which could predict water solubility expressed in logarithm of mols per litre.


**(a)** Coronene (-9.332) **(b)** Acteamide (1.58)

**Figure 3.7.** Least and Most Soluble Molecules of ESOL Dataset. Solubility is reported in the brackets, as –log of solubility data expressed as mols per litre

Figure 3.7 shows the least and most soluble molecule in the ESOL Dataset: coronene and acetamide. It is possible to see that coronene, being symmetrical and being constituted of 7 connected aromatic benzene rings is highly apolar and endowed of very low solubility, on the other hand, acetamide, being formed by just one oxygen, one carbon and one nitrogen is high polar and thus highly soluble.

Ideally, an explanation model, should attribute positive values to the acetamide's atoms and negative values to coronene's atoms.

Figure 3.9 offers another interesting visualization using the 2-undecanol and the tetradecane. the two presented molecules, although seem very similar being formed of a long chain of carbons, have very different values of solubility, due to the presence of the -OH group in the 2-undecanol.

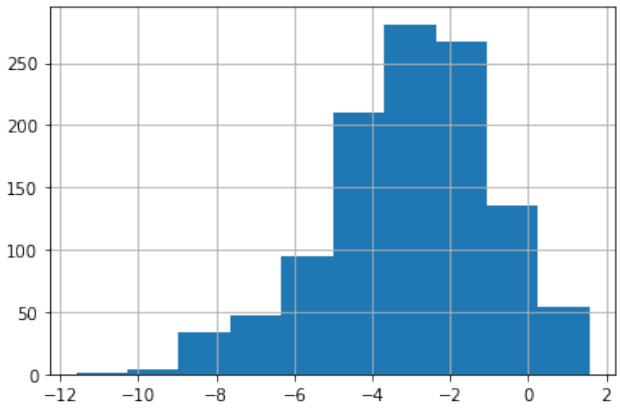



**Figure 3.8.** Distribution of solubility values of ESOL Dataset


**(a)** 2-Undecanol (-2.94) **(b)** Tetradecane (-7.96)

**Figure 3.9.** Comparison of two similar molecules in ESOL Dataset

### 3.2.2 pKa Prediction

$K_a$ is a value which indicates the acidity power of a molecule and is defined as the ratio between the concentration of dissociated hydrogens over the acid concentration present in solution:

$$K_a = \frac{[H^+][A^-]}{[HA]} \tag{3.2}$$

As it happens with most of chemical properties $K_a$ is generally transformed using the $-log$, obtaining the $pK_a$.

Similarly to solubility, it is often possible to infer whether a molecule is more acid than another looking at its structure. This is the main reason why $pK_a$ prediction turns out to be a very good benchmark for XAI in chemistry.

$pK_a$ constitutes an important molecular property to take into account during the drug development process. Acidity and basicity, not only influence the solubility of drugs but, more importantly, they affect absorption, distribution, metabolism, excretion and toxicity (ADMET).

#### Dataset

The source of the data used for approaching this problem is ChEMBL, which is, one of the most populated and used dataset for drug design. ChEMBL contains more than 2 milions of compounds spanning on more than 14 thousands of biological targets and listing over 17 millions of activity data.

ChEMBL, not only stores the activity values of the molecules, but reports also a huge amount of experimental and calculated molecular properties. In particular, this study focuses on the $pK_a$ which ChEMBL calculates using the molcalc module from ChemAxon.

The $pK_a$ values used for the next experiments will be, in fact, predicted values and not measured. This does not constitute a big problem since ChemAxon claims that their model to be very accurate and for the purpose of herein application represent a good reference. In particular the study will focus on the molecules' CX ApKa property, the ChemAxon predicted pKa for the acid functional group. This means that in case of molecules with more than one functional group, only the acid one is considered.

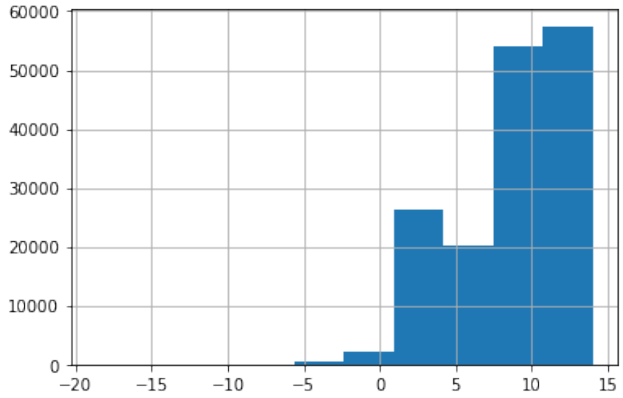
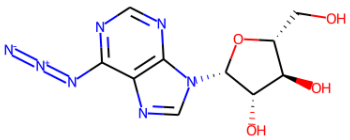
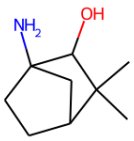



**Figure 3.10.** Distribution of solubility values of pKa ChEMBL Dataset




**(a)** CX ApKa = 14




**(b)** CX ApKa = -18.66

**Figure 3.11.** Least and Most Acid Molecules of pKa ChEMBL Dataset

A dataset of ≈ 160000 molecules with associated values of pKa was downloaded from ChEMBL and was used to train a GCN-QSAR model. Successively, the molecules were perturbed and an explanation model was trained to attribute the atoms' contribution to pKa.

## 3.3 Network Architectures

For each problem described above two models were generated:

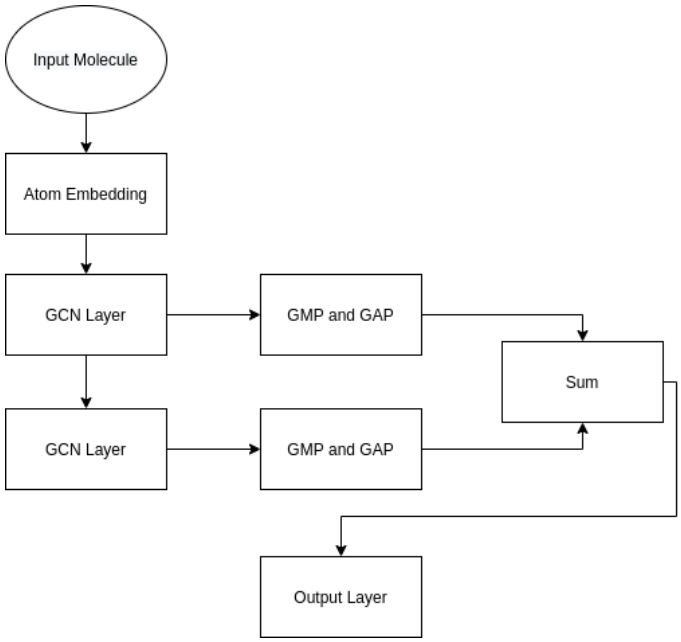
- One GCN-QSAR Model to predict the activity trained on 60% of the datasets and validated on the remaining 40%
- One GGN Explanation Network

Both the networks use Vanilla GCN Layers and are trained using an Adam Optimizer.

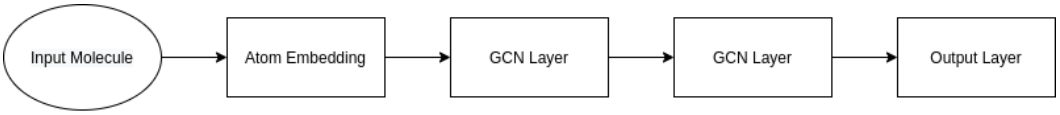
Each network has, at the beginning, one "Atom Embedding Layer", which is a linear layer that takes the input features for each atom, which are a composition of one hot encoded data and floats, and converts it into a latent representation of size 256.

The first architecture's backbone consists in two consecutive GCN layers: the outputs of the two layers are vectorized using global mean and global average pooling, in this way two latent representations for the molecule graphs are obtained. The two vectors are finally summed and fed to a final linear layer which outputs the prediction of the network. For a better understanding, Figure 3.12a shows a block diagram of the first network.

The second network is instead only made of two GCN Layers, since the target of this network will be a graph with the same structure as the input one containing the scores for each atom. This architecture is represented in Figure 3.12b using a block diagram




**(a)** Prediction Network




**(b)** Explanation Network

**Figure 3.12.** Neural Network Architecture Diagrams

# Chapter 4

# Results

This chapter analyses the results obtained in the datasets shown above. Initially, the GCN-QSAR performances will be presented, then an inspection of explainability results will follow and finally the explanations will be used to generate molecules starting from already existing ones trying to optimize some properties.

## 4.1 Graph-QSAR Performances

This section analyses the performances of the GCN-QSAR models trained on the two datasets. Due to the size of the pKa dataset, only the solubility one was used for parameters tuning and the best parameters found were used for both the experiments.

Different combinations of batch size and hidden feature size were tried. The best values found for batch size and hidden feature size are respectively 300 and 256.

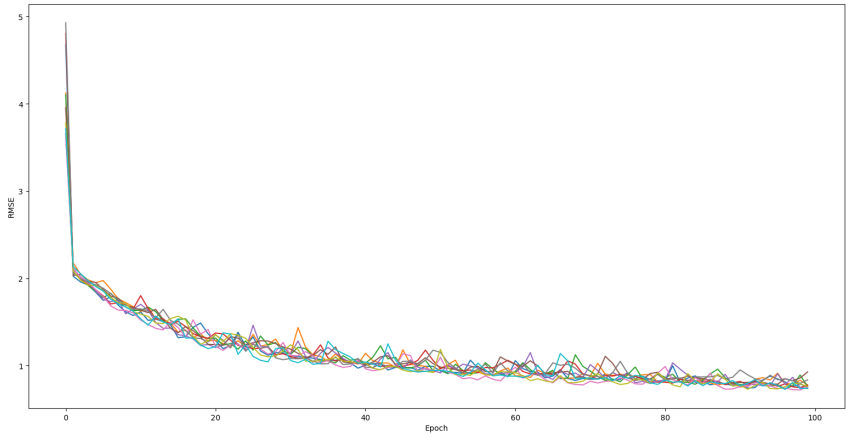



**Figure 4.1.** RMSE evolution over 100 training epochs for the solubility dataset

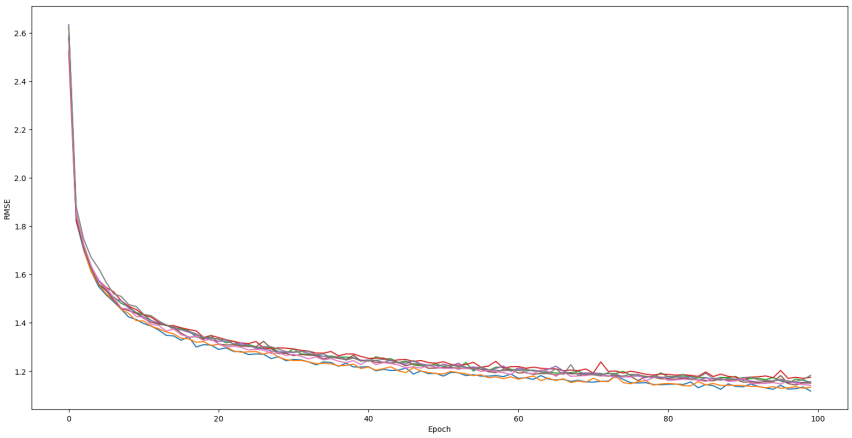



**Figure 4.2.** RMSE evolution over 100 training epochs for the pKa dataset

After choosing the parameters, model's training was performed ten times to evaluate the performances through different tests. Figure 4.1 and 4.2 show the RMSE plots of the training for both solubility and pKa datasets.

Overall the final performances are summarised in Table 4.1 with the mean and standard deviation values for the MSE over the ten runs.

| Model | Validation MSE |
|---|---|
| GCN | $0.734 \pm 0.065$ |
| Linear Regression | $1.542 \pm 0.155$ |
| Random Forest Regressor | $1.441 \pm 0.177$ |

**Table 4.1.** QSAR models performances for the solubility dataset

For comparison purposes, classic machine learning models often used in drug design, Table 4.1 compares results obtained on the solubility dataset with Linear Regression, Random Forest and Support Vector Machine (SVM). This comparison was only performed with the solubility since the size of the pKa dataset was prohibitive for a machine learning approach. It is interesting to notice how GCN performances are comparable to classic machine learning models approaches.

## 4.2 Explainability Results

This section shows the results obtained on the solubility and acidity tasks introduced in the previous chapter. For each problem, a GCN-QSAR model was trained to predict the properties and activities of the molecules. Then, for each model, a GGN Explanation network was trained.

For each test, the attributions using the 3 following methods are compared:

- Integrated gradients
- DeepLIFT
- Perturbation based approach

Attributions are shown using colors over atoms, in particular negative contributions are highlighted using blue while positive ones are highlighted with red. The intensity of the attribution is represented through the opacity of the color: higher values correspond to higher opacity.

In order to have a quantitative approach for evaluating the performances of the different methods for XAI, the sum of the atom solubility contribution between similar molecules are compared. To find deeper connections the correlation score between the true activity of the molecules and the sum of the explanations is also

analysed. It is possible to see that while for solubility, being the easier task, all the method reach high correlation scores, this changes with the acidity, where not all the methods manage give predictions that have an high correlation score with the activity.

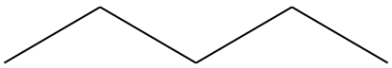
### 4.2.1 Water Solubility

As already mentioned, this problem serves as a benchmark for testing the explanations, due to the fact that generally the solubility of the molecules highly depends on their structure and, in particular, to the presence of certain functional groups.

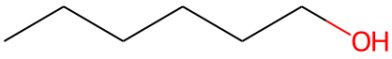
As first analysis, the attributions for 5 different molecules external to the dataset with incremental solubility are compared:

- hexane

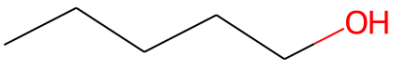
- pentane



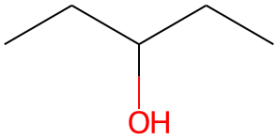
- 1-hexanol




- 1-pentanol




- 3-pentanol



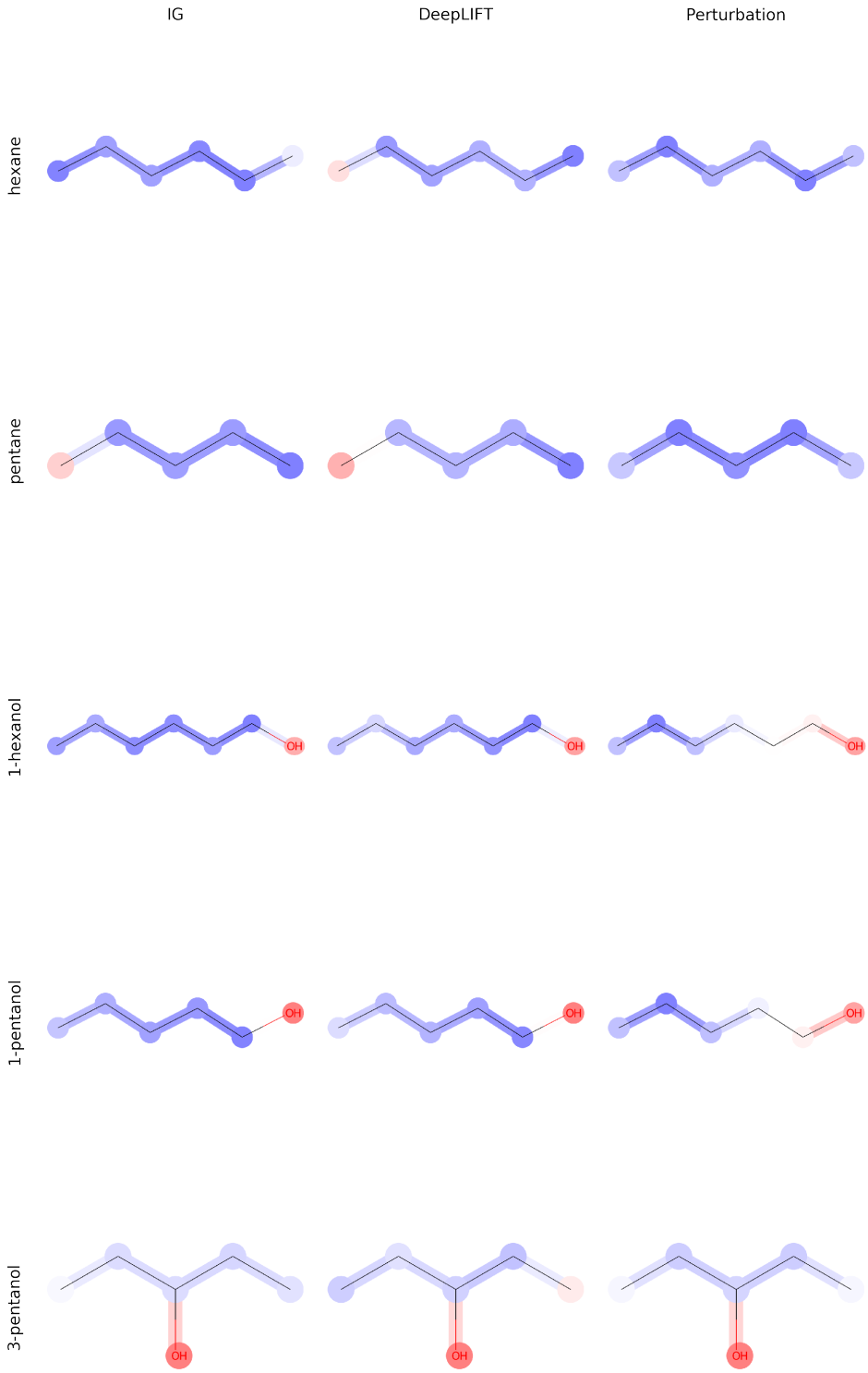



**Figure 4.3.** Attribution comparison varying hydroxyl group position

| Molecule | IG | DeepLIFT | Perturbation | S | logS |
|---|---|---|---|---|---|
| hexane | -0.1166 | -0.1196 | -1.8828 | 0.0001 | -9.1150 |
| pentane | -0.0932 | -0.0956 | -1.8627 | 0.0005 | -7.5483 |
| 1-hexanol | -0.0657 | -0.0711 | -0.5852 | 0.0600 | -2.8134 |
| 1-pentanol | -0.0404 | -0.0460 | -0.5696 | 0.2500 | -1.3863 |
| 3-pentanol | -0.0096 | -0.0163 | -0.1369 | 0.5800 | -0.5447 |

**Table 4.2.** Explanations and experimental solubility values

Chemical theory and experimental values (Table 4.2) for solubility of these molecules suggest that the hydroxyl ($OH^-$) group generally increases the solubility if introduced in a molecule. Moreover, its position has an important role in modulating the solubility: the more the hydroxyl group is at the center of the carbon chain, the more the solubility increases.

The hexane and the pentane are only composed of a chain of carbons, which render them apolar and highly insoluble in water. Adding a hydroxyl group in hexane and pentane position 1 the resulting 1-hexanol and 1-pentanol are generated respectively. Both 1-hexanol and 1-pentanol are more soluble than their cognate molecules and in fact the hydroxyl group receives positive scores from all the considered XAI methods. Finally, moving the hydroxyl group in position 3, further increments the solubility. In addition, in this case the three methods are in agreement with the experimental values (Figure 4.3): the $OH^-$ group is attributed much more positively in the 3-pentanol compared to the 1-pentanol.

One more result that is interesting is that with the proposed attribution method all the pentane's carbons receive negative scores, while for the other methods, the last carbon is attributed positively. This does not make any sense in terms of chemistry since the first and second carbons should be attributed equally also because the molecule is perfectly specular.

Analysing the sum of the attributions (Figure 4.4, Table 4.2), over all, all the methods are able to correctly explain the molecules' solubility. In fact, a high positive correlation was observed between IG, DeepLIFT, Perturbation and the log solubility. Although all the methods behave correctly, the herein proposed approach showed the highest correlation score with the log solubility.

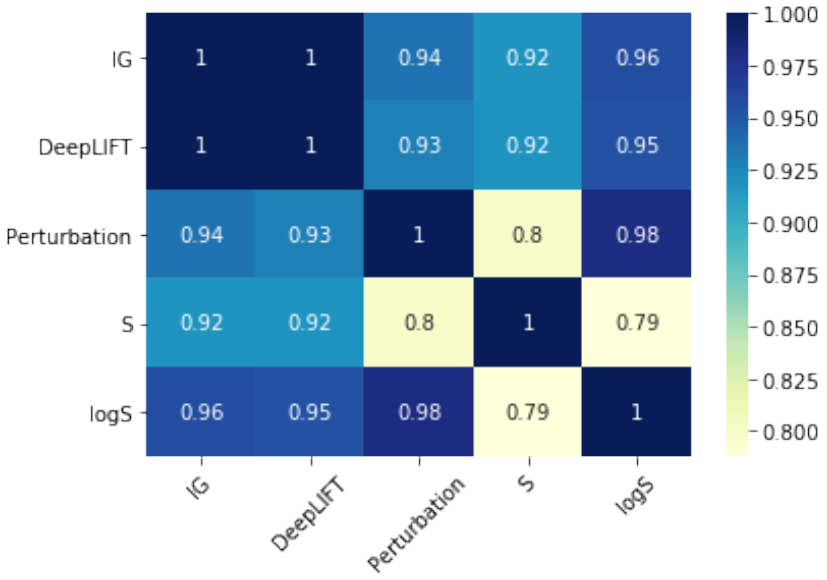



**Figure 4.4.** Correlation matrix between sum of attributions and solubiltity

### 4.2.2 pKa

pKa is a property highly dependent on the structure of the molecules due to various molecular aspects such as steric hindrance, atoms electronegativity and resonance.

Since pKa can be influenced by different chemical and physical effects the perturbation based approach was tested in two experiments:

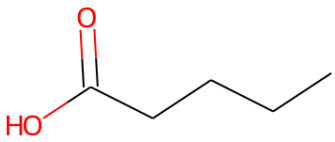
- 5 molecules derived by the pentanoic acid: this test compares the pKa among 5 molecules with the same alifatic scaffold to which the chlorine has been added in different positions.
- 5 molecules derived by the phenol: this test compares the pKa among 5 molecules with the same aromatic scaffold to which the $NO_2$ group has been added in different positions.

#### Pentanoic Acid Experiment

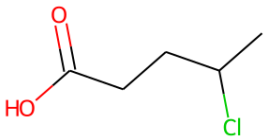
For this experiment, the attributions for five molecules with increased acidity (decreased pKa) among each other have been compared. In particular, due to the inductive effect, adding chlorine to the pentanoic acid its pKa decreases. Moreover, the pKa decreasing is enhanced by the proximity of the chlorine atom to the carboxyl group (inductive effect).

The molecules taken in exam were the following:

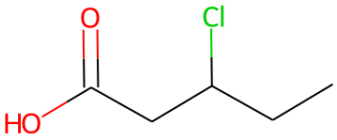
- pentanoic acid



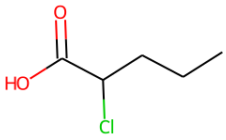

- 4-chloropentanoic acid



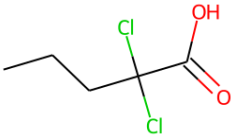

- 3-chloropentanoic acid




- 2-chloropentanoic acid




- 2,2-dichloropentanoic acid



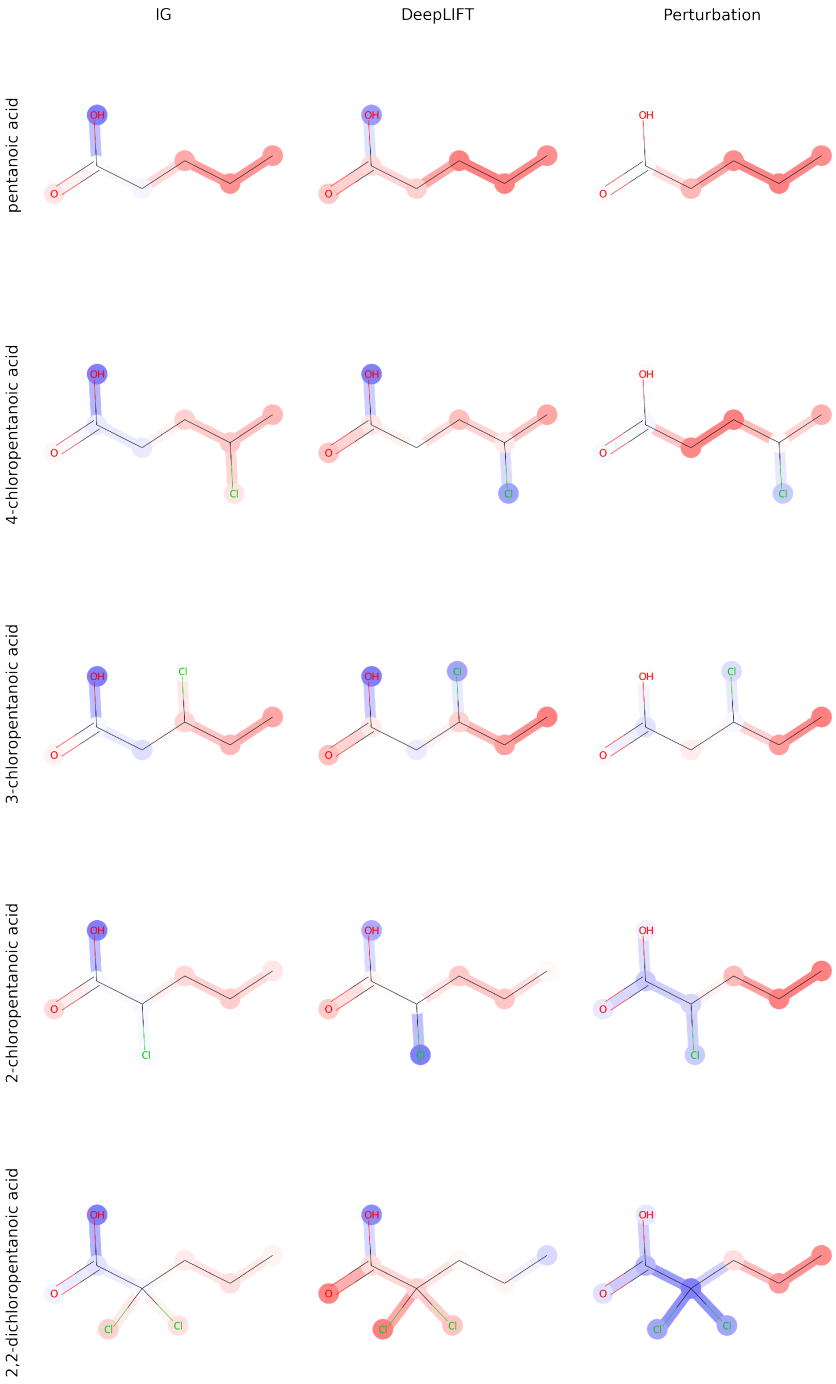



**Figure 4.5.** Attribution comparison adding chlorine in different positions of the pentanoic acid

Looking at the explanations (Figure 4.5) generated using the perturbation based approach it is possible to observe that for the pentanoic acid, the carbon chain is considered to be part of the molecule leading to pKa increments (less acid), while the carboxyl group receives the lowest score, since it is recognized as the group directly responsible for the acidic behaviour.

Adding a chlorine in position 4 decreases the pKa and it was correctly attributed of a negative score by the proposed method.

Shifting the chlorine atom to position 3 and 2 lead to a further pKa decrement which chemists explain with the inductive effect. The perturbation based approach correctly scored negatively the chlorine increasing the intensity when it is closer to the carboxyl group. Finally the 2,2-dichloropentanoic acid, which is the most acidic molecule among the set, received the lowest negative scores on both the chlorines and the carboxyl moieties.

Analysis of the other methods results, did not lead to individuate any pattern able to clearly explain the chlorine effect on the acidity. Moreover, both IG and DeepLIFT incorrectly attributed positively the chlorines in the 2,2-dichloropentanoic and 2-chloropentanoic acid, respectively, as if, they would be responsible for a pKa increment. Again, the perturbation based approach proved to be effective in explaining the properties of the molecule attributing correctly the atoms which mainly influence them.

| Molecule | IG | DeepLIFT | Perturbation | pKa |
|---|---|---|---|---|
| pentanoic acid | 0.4884 | 0.9821 | 1.6650 | 5.01 |
| 4-chloropentanoic acid | 0.1337 | 0.0765 | 0.5025 | 4.27 |
| 3-chloropentanoic acid | 0.1610 | 0.2180 | 0.5006 | 4.24 |
| 2-chloropentanoic acid | 0.0253 | -0.0779 | 0.4777 | 3.96 |
| 2,2-dichloropentanoic acid | -0.1603 | 0.4864 | -0.7181 | 3.19 |

**Table 4.3.** Explanations and experimental pKa values for the pentanoic acid experiment

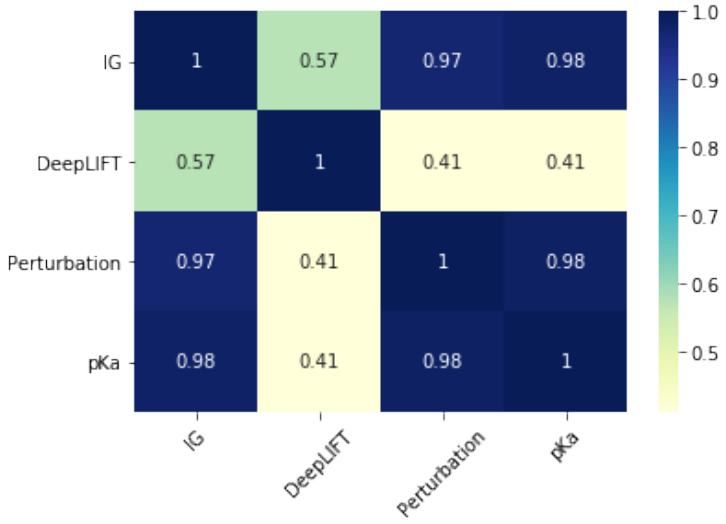



**Figure 4.6.** Attribution comparison varying hydroxyl group position

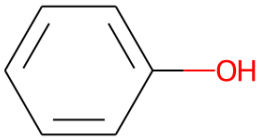
Correlation scores (Figure 4.6, Table 4.3) between the sum of attributions and experimental pKa values confirm the fact that the perturbation based approach performs better than DeepLIFT for this case. Moreover, analysing at the visual saliency maps it was possible to see that IG, actually, wrongly attributed positively the chlorine.

### Nitrophenols Experiment

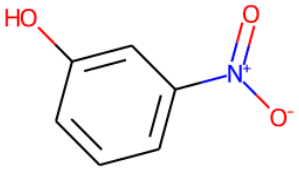
In this section, the explanations for 5 molecules derived by adding the $NO_2$ group to different positions of a phenol are discussed. It is known that the $NO_2$ group increases the acidity of the phenolic hydroxyl through stabilization of its conjugated base by means of resonance and inductive effects.

The position of the substituent highly influences the pKa, in particular there is a decreasing trend between the following molecules:

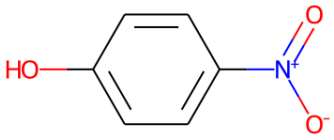
- phenol



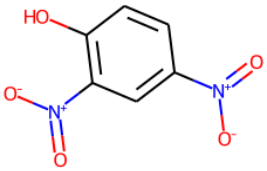

- 3-nitrophenol




- 4-nitrophenol




- 2,4-dinitrophenol



- 2,4,6-trinitrophenol

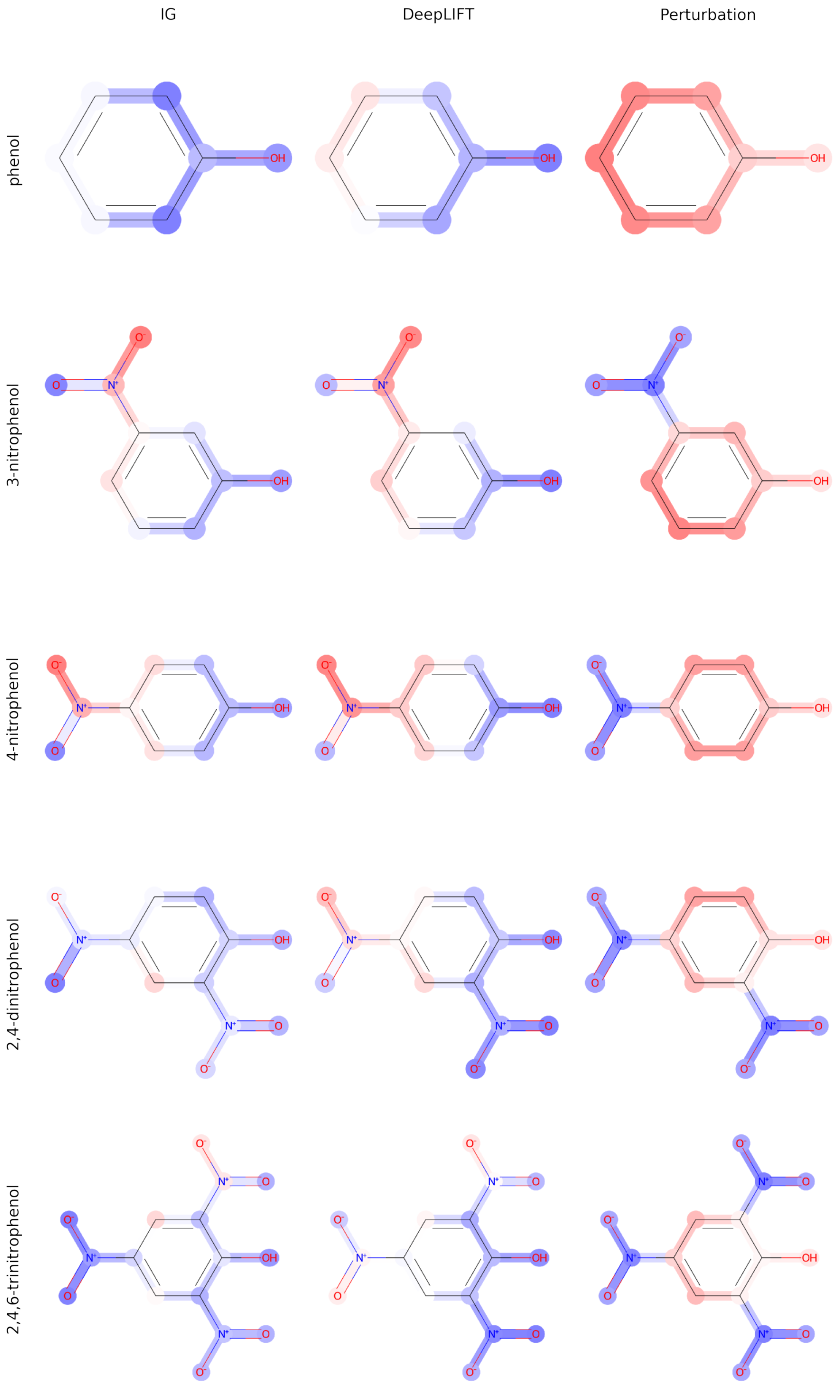



**Figure 4.7.** Attribution comparison between nitrophenols

Table 4.4 summarizes the pKa values and attributions for these molecules, while

Figure 4.7 shows the molecules together with atom attributions comparing IG, DeepLIFT and the perturbation based approach.

Analysing the results obtained with the latter approach it is evident how the $NO_2$ group is attributed responsible for increasing of acidity. It is interesting to see how the perturbation based approach is able to attribute the whole $NO_2$ group while IG and DeepLIFT do not consider the group as a whole.

The perturbation based approach is the only one which manages to attribute the molecules reaching high correlation scores between the sum of explanations and the experimental pKa value (Figure 4.8). In particular, the method seems to be able to correctly assign the different importance to the mesomeric of the inductive effect. in the case of 3-nitrophenol the $NO_2$ group is manly involved in an inductive effect to stabilize the phenolate ion (phenol conjugate base). Differently, for the 4-nitrophenol the conjugation effect (mesomeric) is mainly responsible for the phenolate stability. Although only with a slightly extend 3-nitrophenol was correctly predicted more acidic than 4-nitrophenol by the perturbation method. On the contrary, IG and DeepLIFT methods were not able to reproduce the correct acidity trend, indicating the phenol to be more acidic than either 3-nitrophenol or 4-nitrophenol.

| **Molecule** | **IG** | **DeepLIFT** | **Perturbation** | **pKa** |
|---|---|---|---|---|
| phenol | -1.2193 | -1.6356 | 2.4448 | 10.02 |
| 3-nitrophenol | -0.7874 | -0.3438 | 0.7564 | 7.89 |
| 4-nitrophenol | -0.6525 | -0.1240 | 0.7420 | 7.07 |
| 2,4-dinitrophenol | -2.6882 | -2.8814 | -0.9276 | 4.50 |
| 2,4,6-trinitrophenol | -3.9940 | -5.1985 | -2.5918 | 2.34 |

**Table 4.4.** Phenol derivatives attribution and pKa values

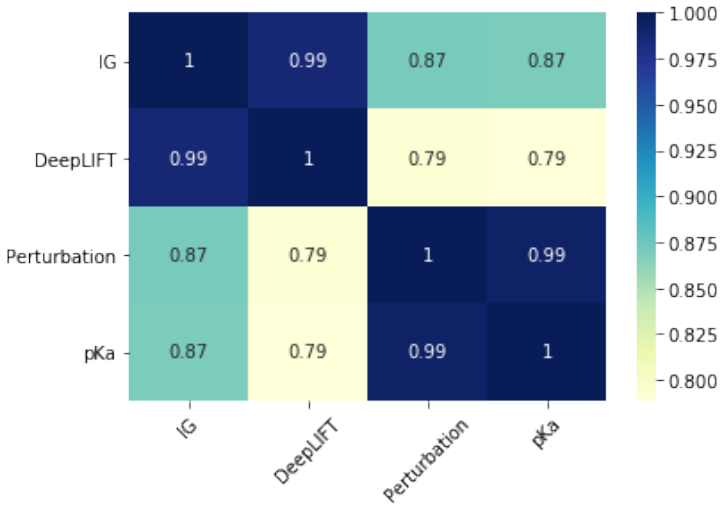



**Figure 4.8.** Attribution correlation scores comparison varying $NO_2$ group position

## 4.3 Molecule Optimization with XAI

As mentioned in the previous section, one of the main uses of XAI in drug design is its employ to suggest molecular modifications to increase their biological activity. This is called drug optimization and could be performed automatically by adding or substituting single atoms or groups with others.

To test the generation of new optimized molecules, the three explanation methods shown in the previous chapters have been used to determine the part of the molecules which most influence the molecular properties in order to suggested some targeted modifications.

In this chapter, for both the previously cited datasets, some experiments were performed trying to optimize some molecules with the different XAI methods. The molecules have been generated using the same algorithm proposed to generate the couples for training the perturbation based attribution method. In particular, starting from a molecule, 420 new molecules were generated setting $N = 20$ and $D = 2$.

The new generated molecules have been then fed to the GCN-QSAR network in order to predict the new activity. The distributions were then compared to evaluate the performances.

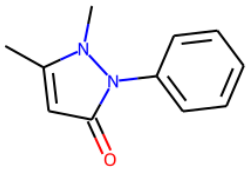
While the initial generation algorithm was purely random, in order to apply targeted modifications the atom attributions have been used as weights for the selection of the atoms to be substituted. In this way the algorithm stays random in the selection of the functional group but will tend to substitute atoms which have lower attributions.

For each experiment, the performances of generation using IG, DeepLIFT and the Perturbation based approach are analysed comparing the mean value, standard deviation and maximum/minimum value. For each result, also 420 molecules using the purely random algorithm are generated and the obtained activity distribution is compared with the other methods.

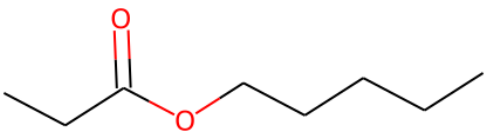
### 4.3.1 Water Solubility

This section analyses the results obtained with the solubility dataset. In particular, the generation algorithm was tested starting from two molecules:

- A very soluble molecule:Phenazone or antipyrene (0.715)




- A molecule with intermediate solubility: Ethyl pentanoate (-1.28)




Since the least and most soluble molecule do not actually have a very big and articulate structure, some bigger molecules were needed in order to have a better understanding of the generation process. Figure 4.9 shows the attributions of the two molecules using IG, DeepLIFT and the perturbation based approach: all the methods more or less agree on the attributions for ethyl pentanoate while regarding antipyrene, there are some variations between the three methods.

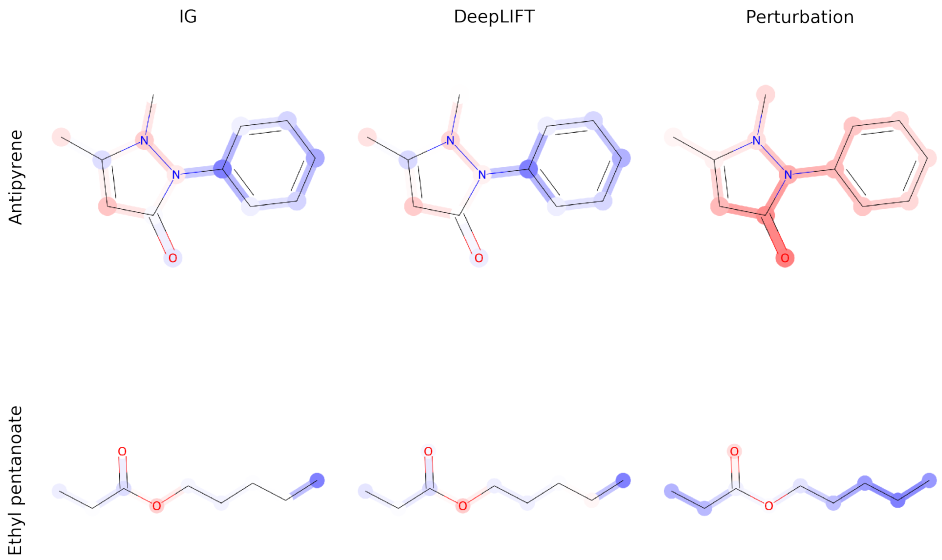



**Figure 4.9.** Solubility attributions for antipyrene and ethyl pentanoate

Figure 4.10 shows the distribution of predicted solubility for the generated molecules starting from phenazone using the different approaches: the molecules generated using XAI are distributed on higher values of solubility. In particular, the perturbation based approach managed to generate the distribution of molecules with the highest mean and maximum values (Table 4.5). All the three distributions reported similar standard deviation values, this was expected since the choice of the functional group is still random.

| Method | Mean | Std | Max |
|---|---|---|---|
| Random | -3.09 | 0.57 | -1.77 |
| Perturbation | -2.67 | 0.58 | -1.08 |
| IG | -2.91 | 0.58 | -1.40 |
| DeepLIFT | -2.73 | 0.56 | -1.30 |

**Table 4.5.** Predicted solubility statistics of molecules generated from antipyrene

Figures 4.11, 4.12, 4.14, 4.13 represent some of the generated molecules using the random approach, the perturbation based method, DeepLIFT and IG respectively. While DeepLIFT and IG consider the aromatic ring to be the part which less

contributes to the solubility, the perturbation based approach attributes the methyl group attached to the 5-atoms ring to influence negatively the solubility, in fact, most of the generated molecules using the perturbation based approach tend to modify the 5-atoms ring while IG and DeepLIFT act more on the benzene ring.

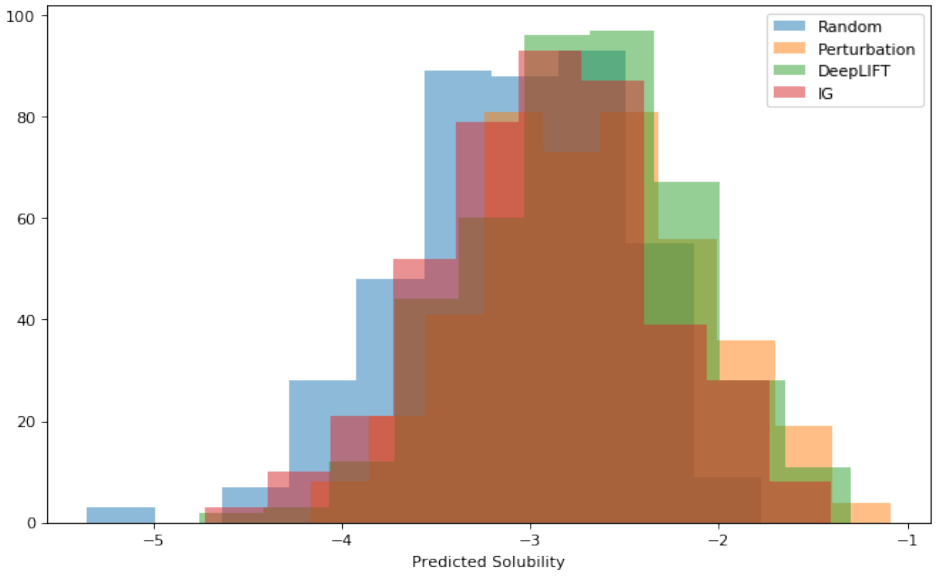



**Figure 4.10.** Predicted solubility distribution of molecules generated from antipyrene

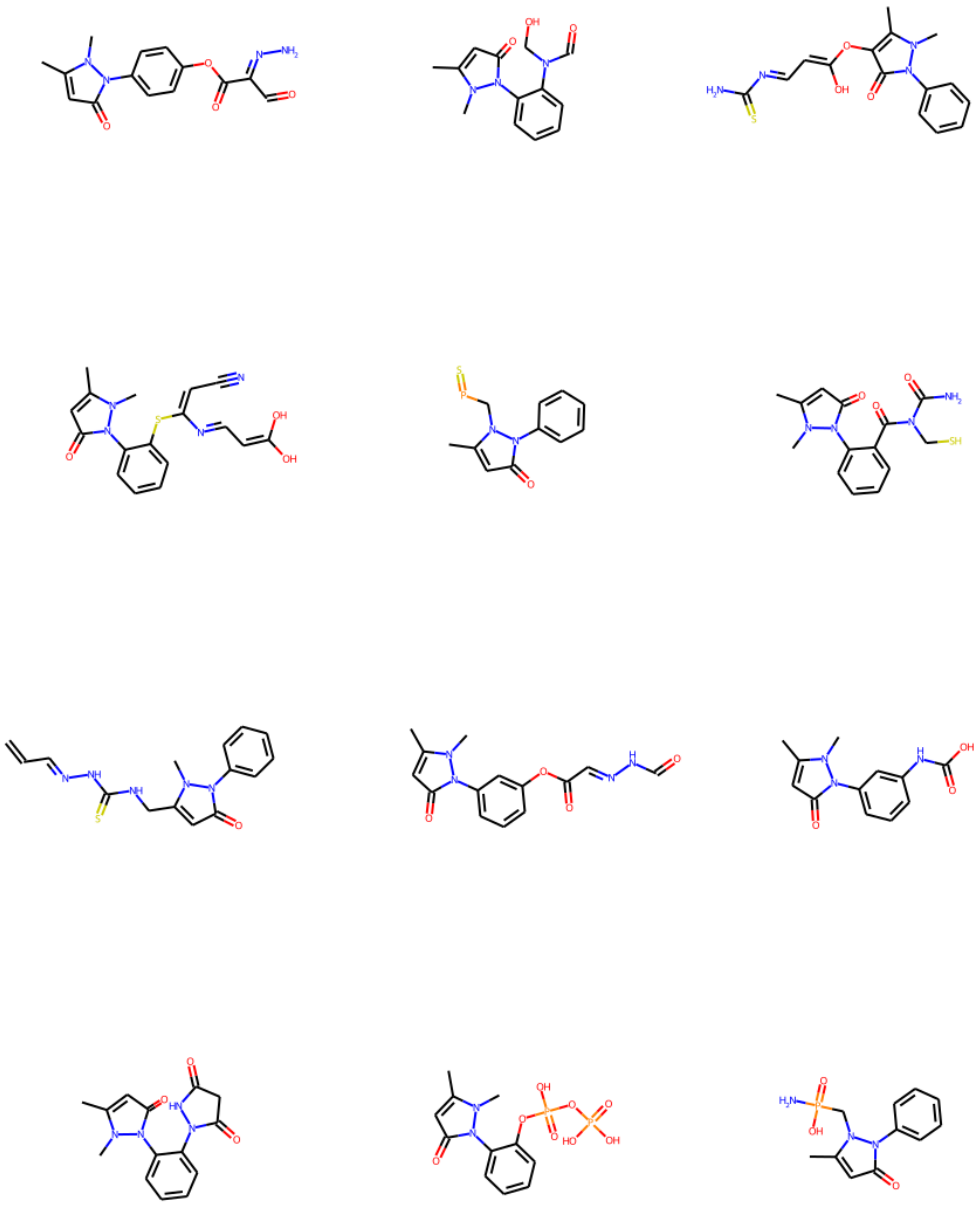


**Figure 4.11.** Molecules generated from antipyrene with random approach

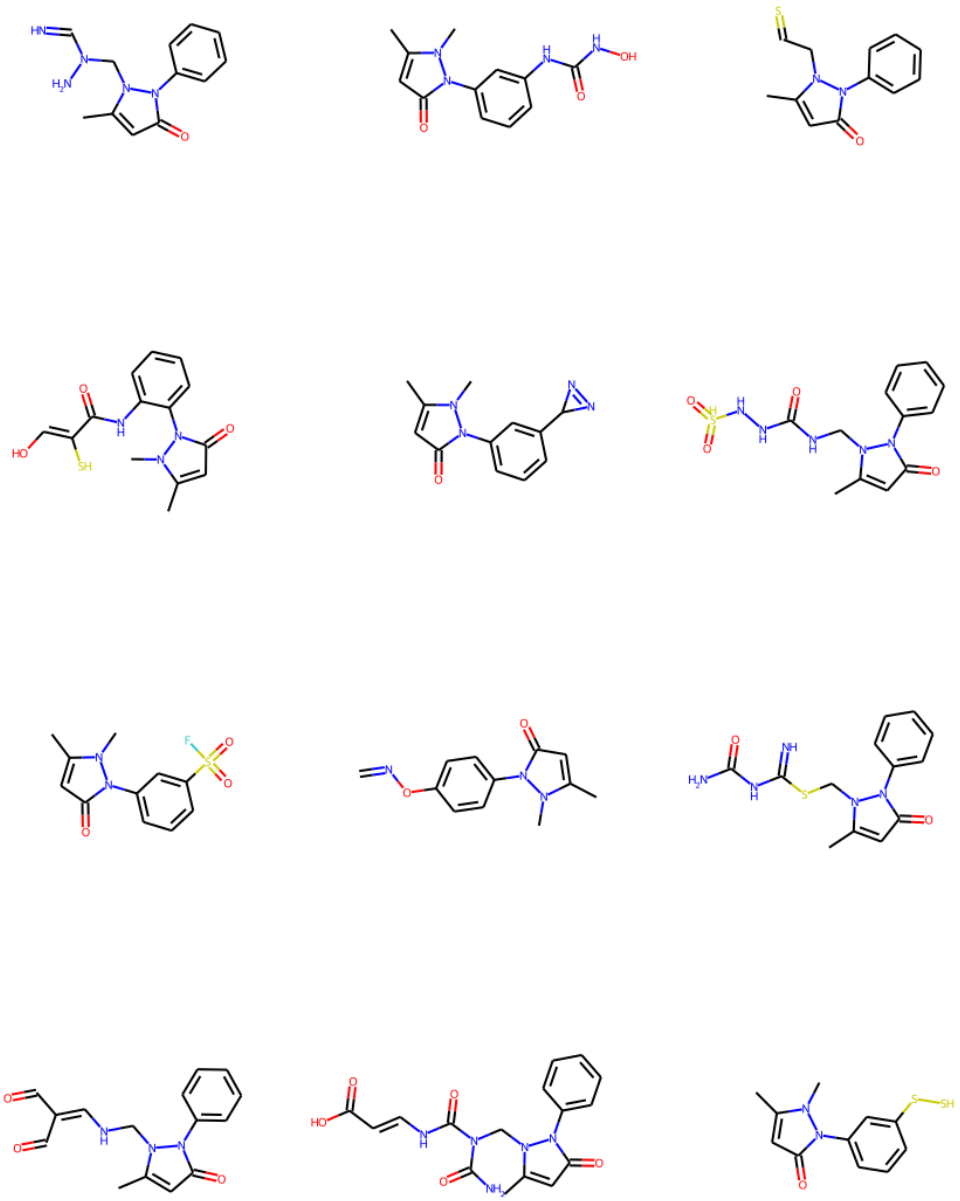


**Figure 4.12.** Molecules generated from antipyrene with perturbation based approach

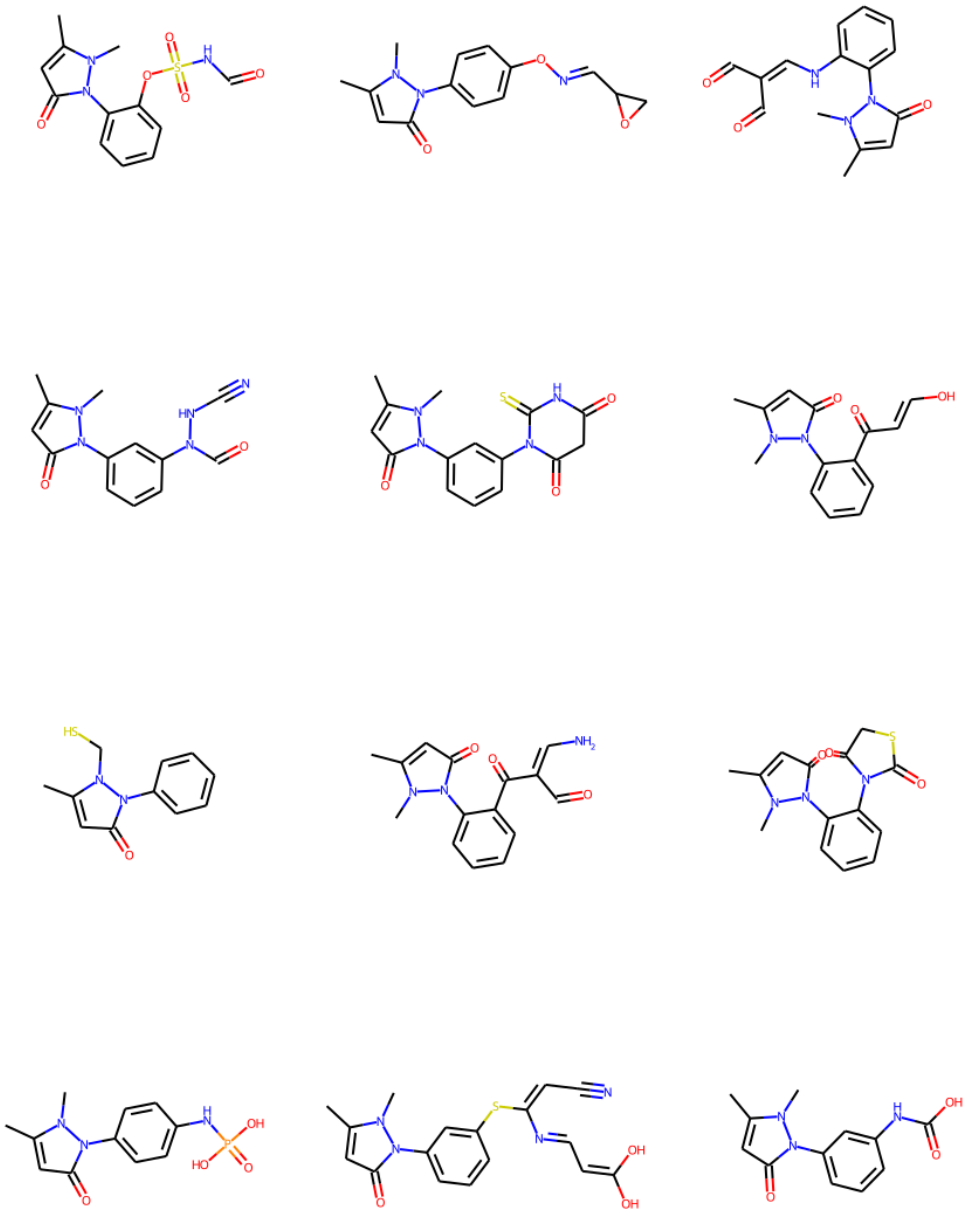


**Figure 4.13.** Molecules generated from antipyrene with IG

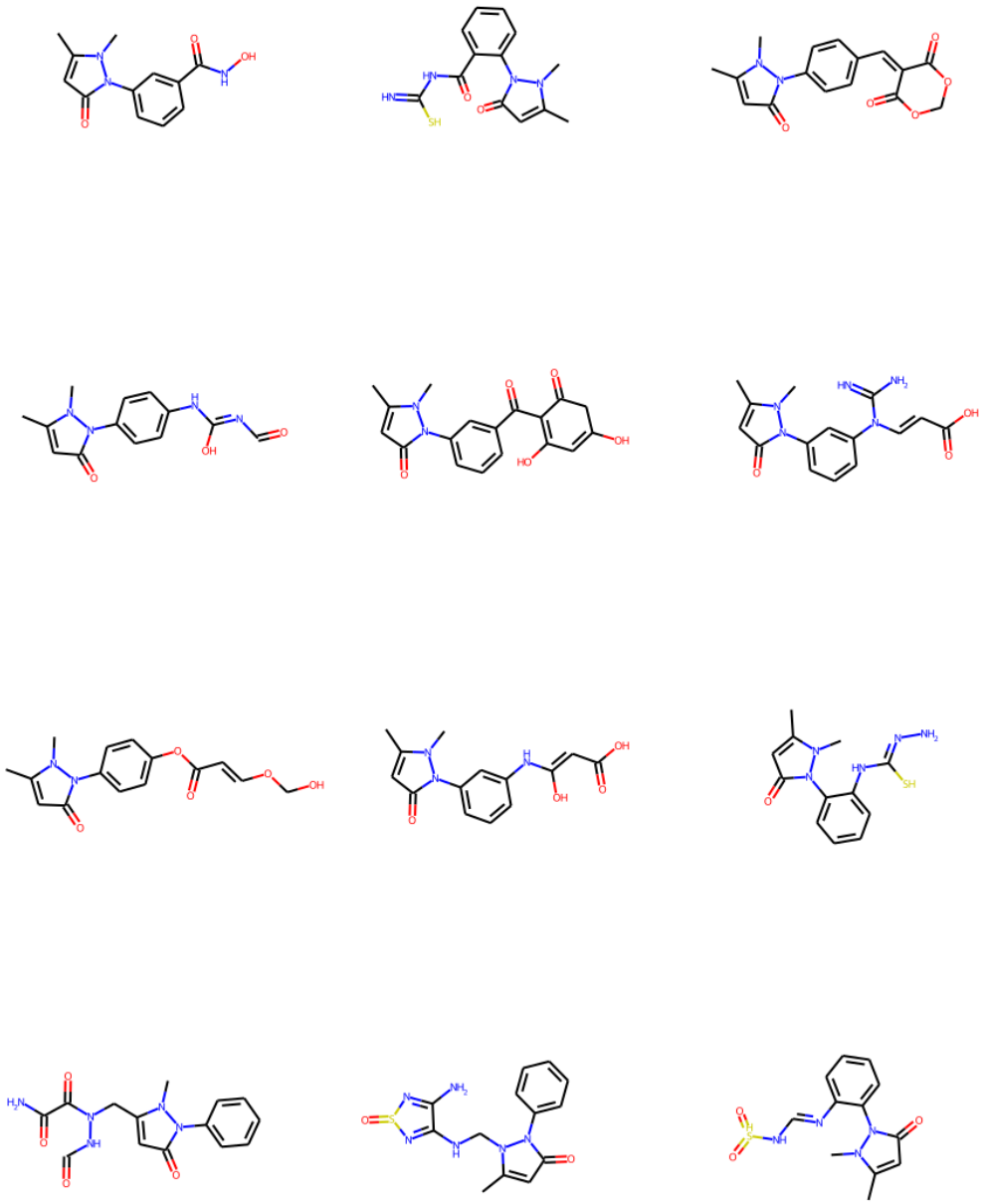


**Figure 4.14.** Molecules generated from antipyrene with DeepLIFT

Similarly to the antipyrene case, for the ethyl pentanoate, the perturbation based approach manages to generate the molecules which are distributed with the highest mean value of predicted solubility (Table 4.6), while the most predicted soluble molecule was generated using DeepLIFT.

It is important to notice that while developing new drugs, generally, it is preferable to generate a set of molecules with higher activity rather than just one. This is due to the fact that it is difficult that the only generated molecule would really result in a possible drug.

| Method | Mean | Std | Max |
|---|---|---|---|
| Random | -2.88 | 0.73 | -0.95 |
| Perturbation | -2.41 | 0.74 | -0.46 |
| IG | -2.78 | 0.90 | -0.74 |
| DeepLIFT | -2.49 | 0.80 | -0.22 |

**Table 4.6.** Predicted solubility statistics of molecules generated from ethyl pentanoate

Figure 4.15 shows the plotted distributions for the generated molecules using the four methods while Figures 4.16, 4.17, 4.19 and 4.18 show some sample molecules for each method.

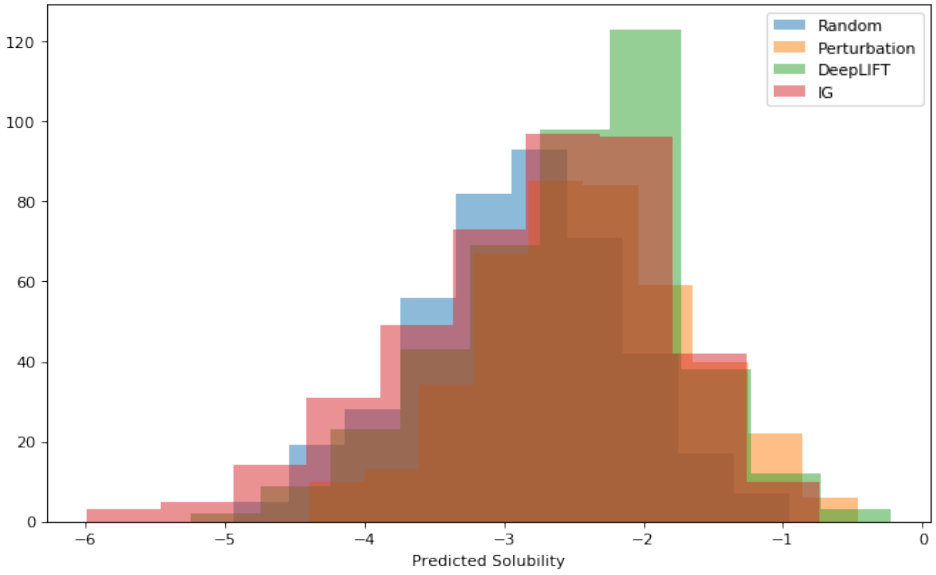



**Figure 4.15.** Predicted solubility distribution of molecules generated from ethyl pentanoate

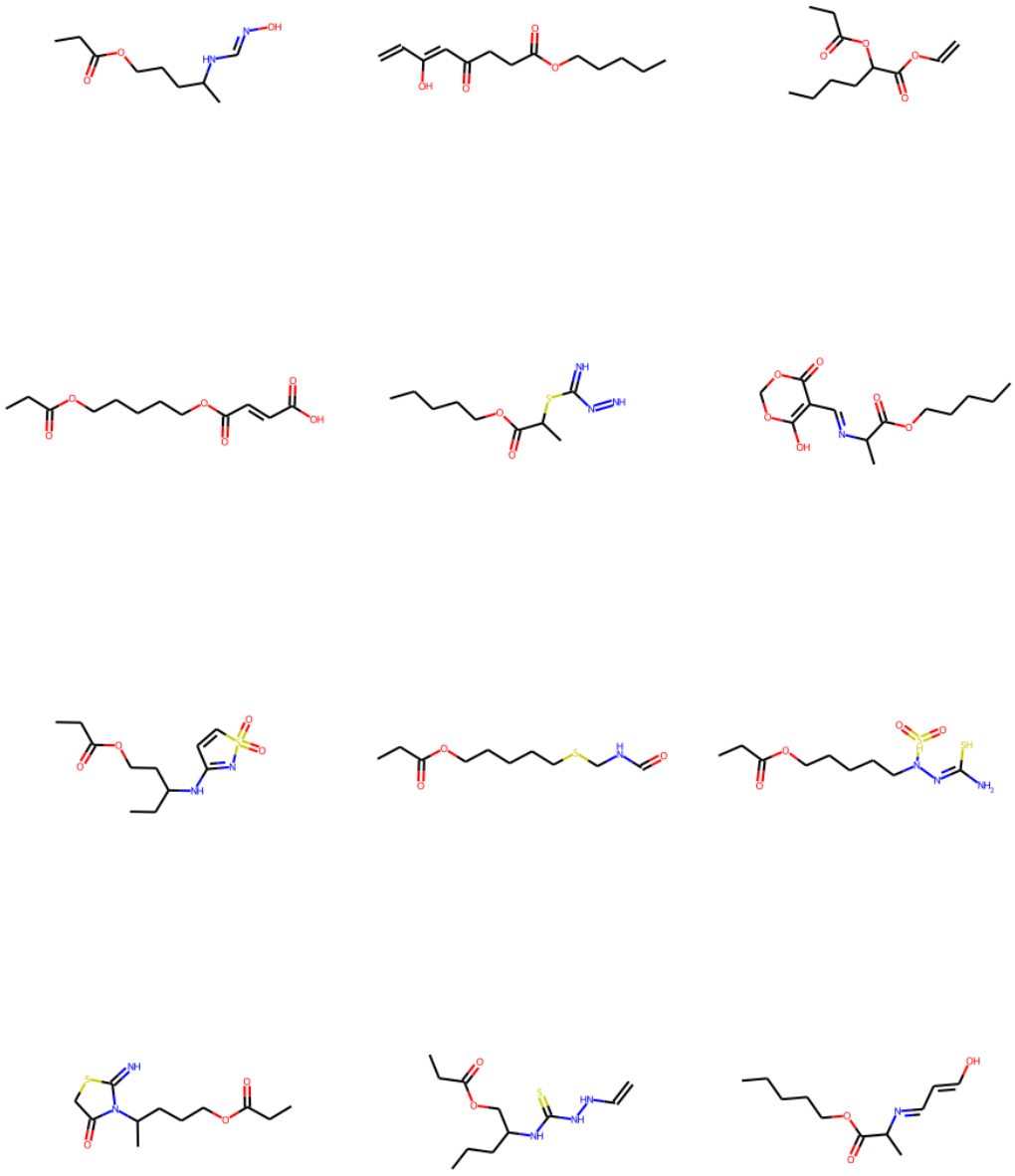


**Figure 4.16.** Molecules generated from ethyl pentanoate with random approach

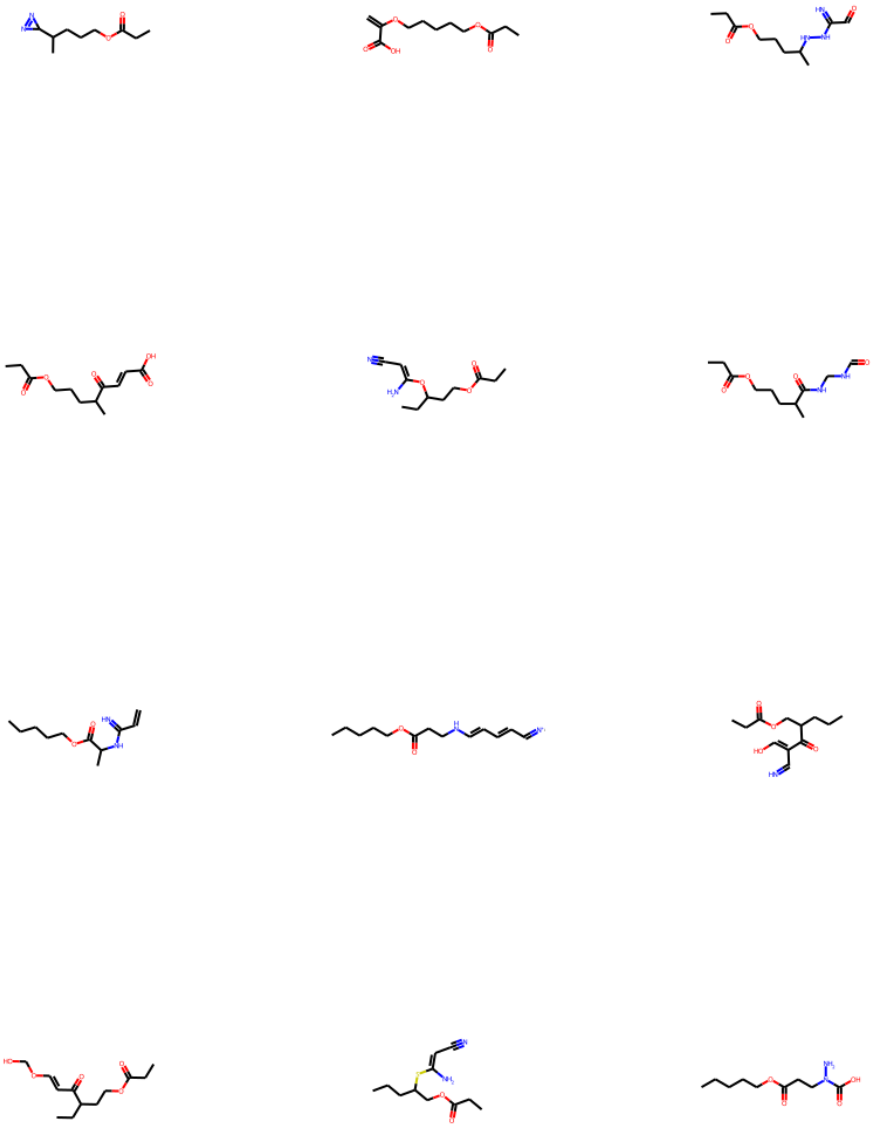


**Figure 4.17.** Molecules generated from ethyl pentanoate with perturbation based approach

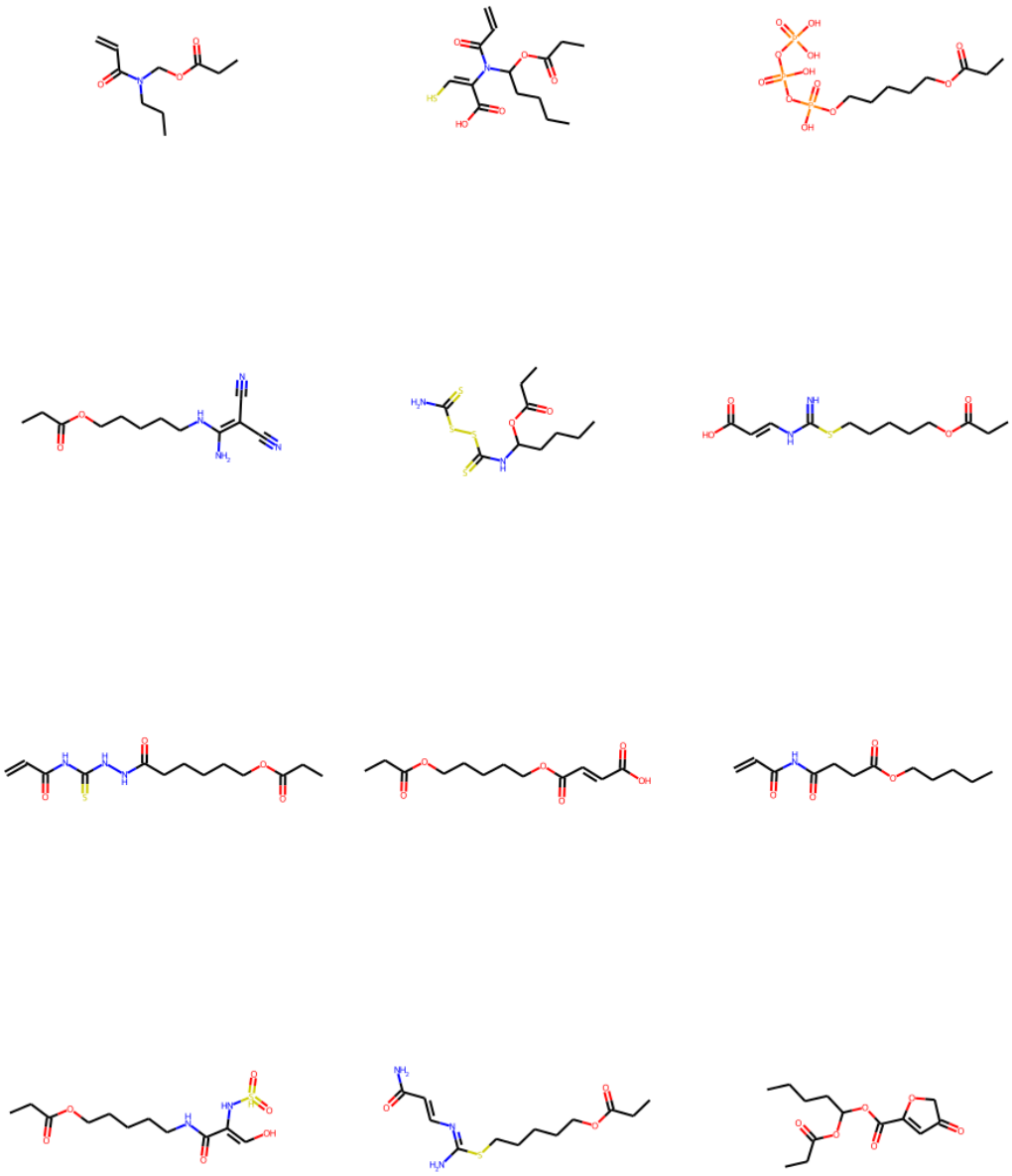


**Figure 4.18.** Molecules generated from ethyl pentanoate with IG

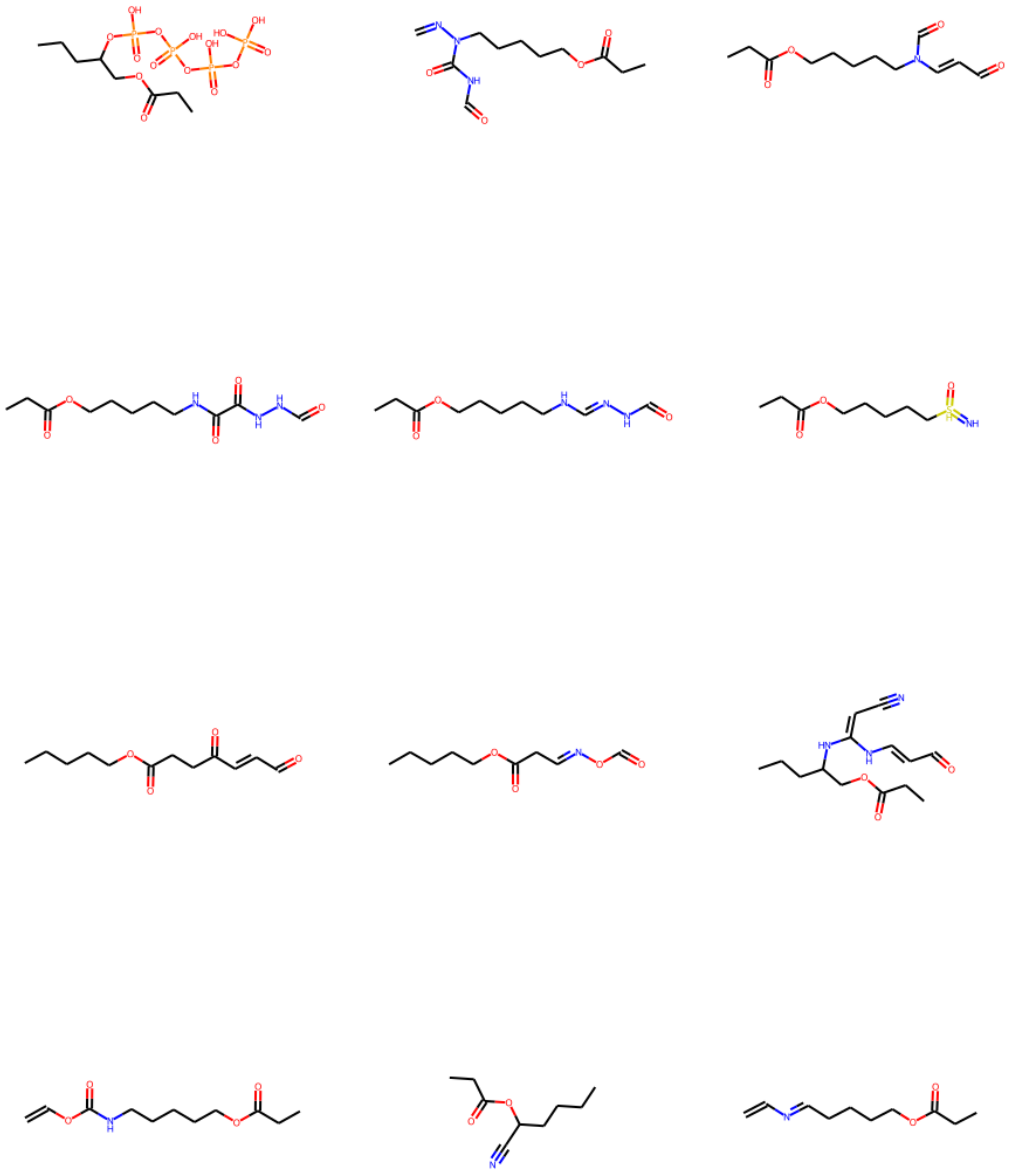


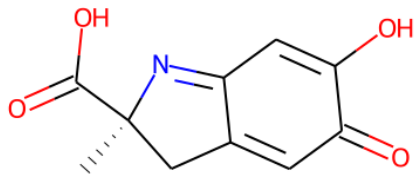
**Figure 4.19.** Molecules generated from ethyl pentanoate with DeepLIFT

### 4.3.2 pKa

While with the solubility experiments the aim was to increase the solubility of the molecules, this section will focus on decreasing the pKa value (increasing the acidity).

For this purpose, the analysis has been performed on the following molecules:

- Molecule 1 (3.6)




- Molecule 2 (8.91)

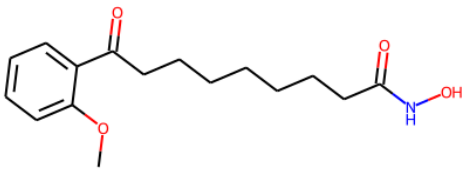



Figure 4.20 shows the attributions for the two molecules using IG, DeepLIFT and the perturbation based approach. In addition, in this case the methods showed to be in a good agreement although some slight differences.

For both molecules (Table 4.7 and 4.26), optimization through perturbation based approach results in distributions which have the lowest mean values compared to the other methods. Although the mean value is the lowest, in both cases DeepLIFT manages to generate the most acid molecule.

For the first molecule, DeepLIFT results might seem better at a first analysis but considering the standard deviation it is possible to see that the perturbation based approach scores a quite lower standard deviation performing “more targeted” modifications.

Figures 4.21 and 4.26 are the plots for the distributions of the predicted pKa values.

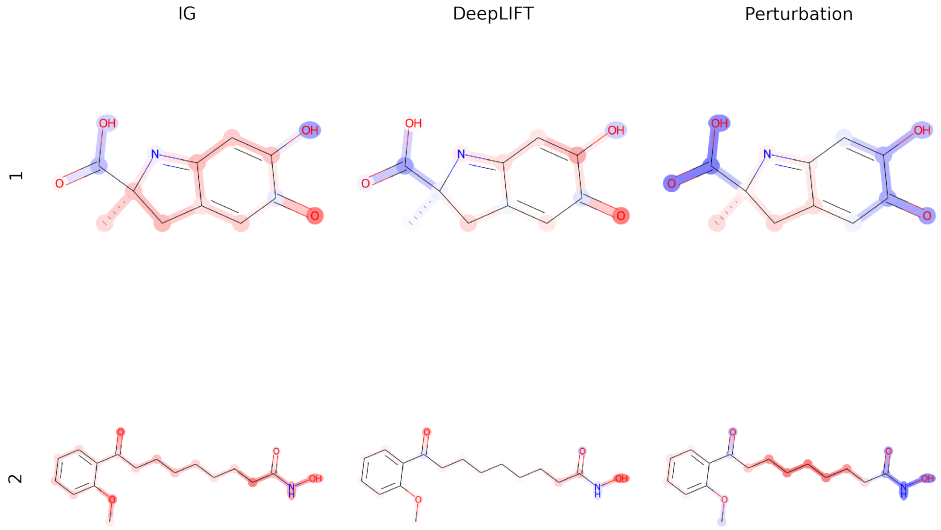



**Figure 4.20.** pKa attributions for the two molecules

Figure 4.22, 4.23 and 4.25, 4.24 show some samples of molecules generated using the random method, the perturbation based approach, DeepLIFT and IG respectively starting from the first molecule.

The same results for the second molecule can be analysed in Figure 4.27, 4.28 and 4.30, 4.29.

| Method | Mean | Std | Min |
|---|---|---|---|
| Random | 4.29 | 1.74 | -1.77 |
| Perturbation | 3.75 | 1.31 | -1.48 |
| IG | 3.86 | 1.48 | -1.85 |
| DeepLIFT | 4.02 | 2.14 | -3.66 |

**Table 4.7.** Predicted pKa statistics of molecules generated from the first molecule

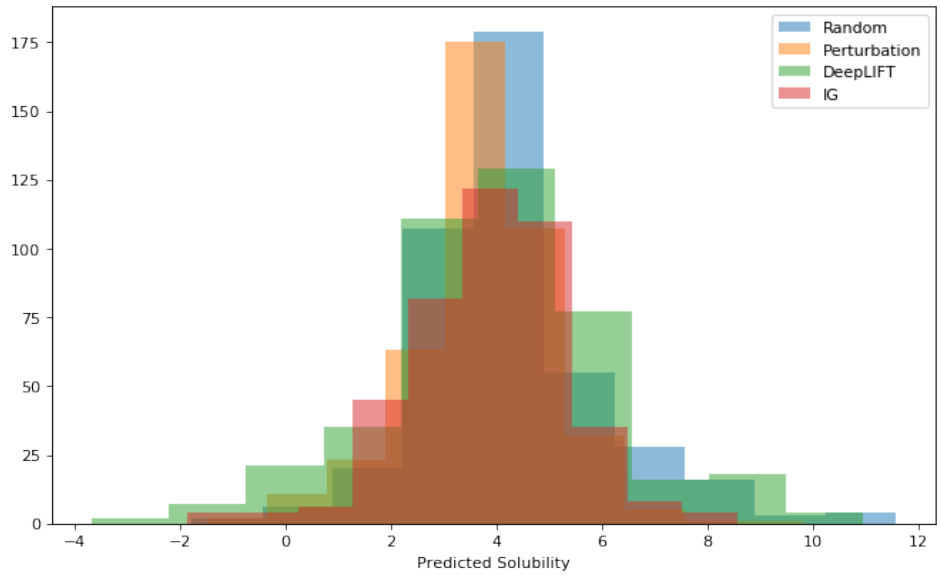



**Figure 4.21.** Predicted pKa distribution of molecules generated from the first molecule

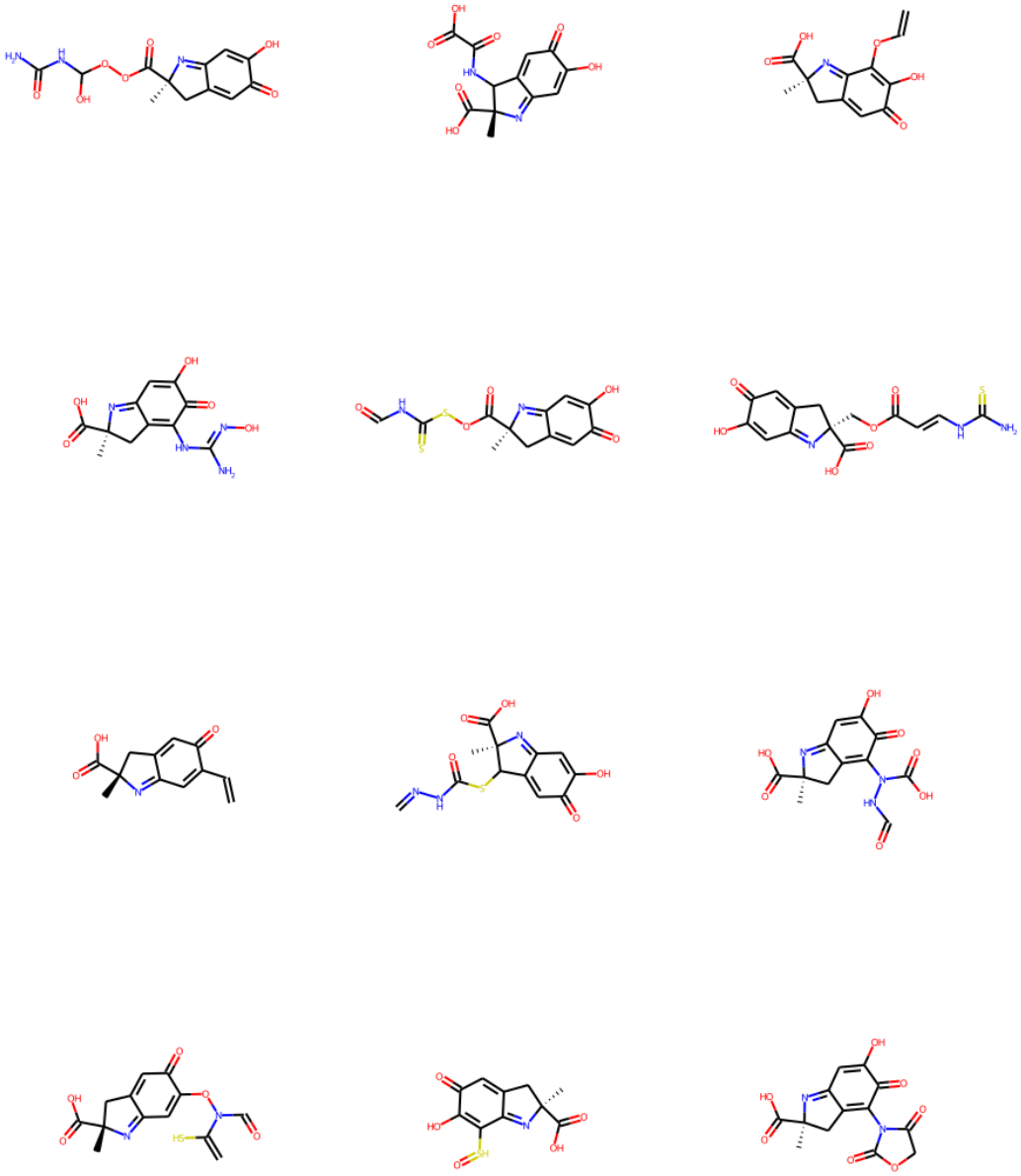


**Figure 4.22.** Molecules generated from the first molecule with random approach

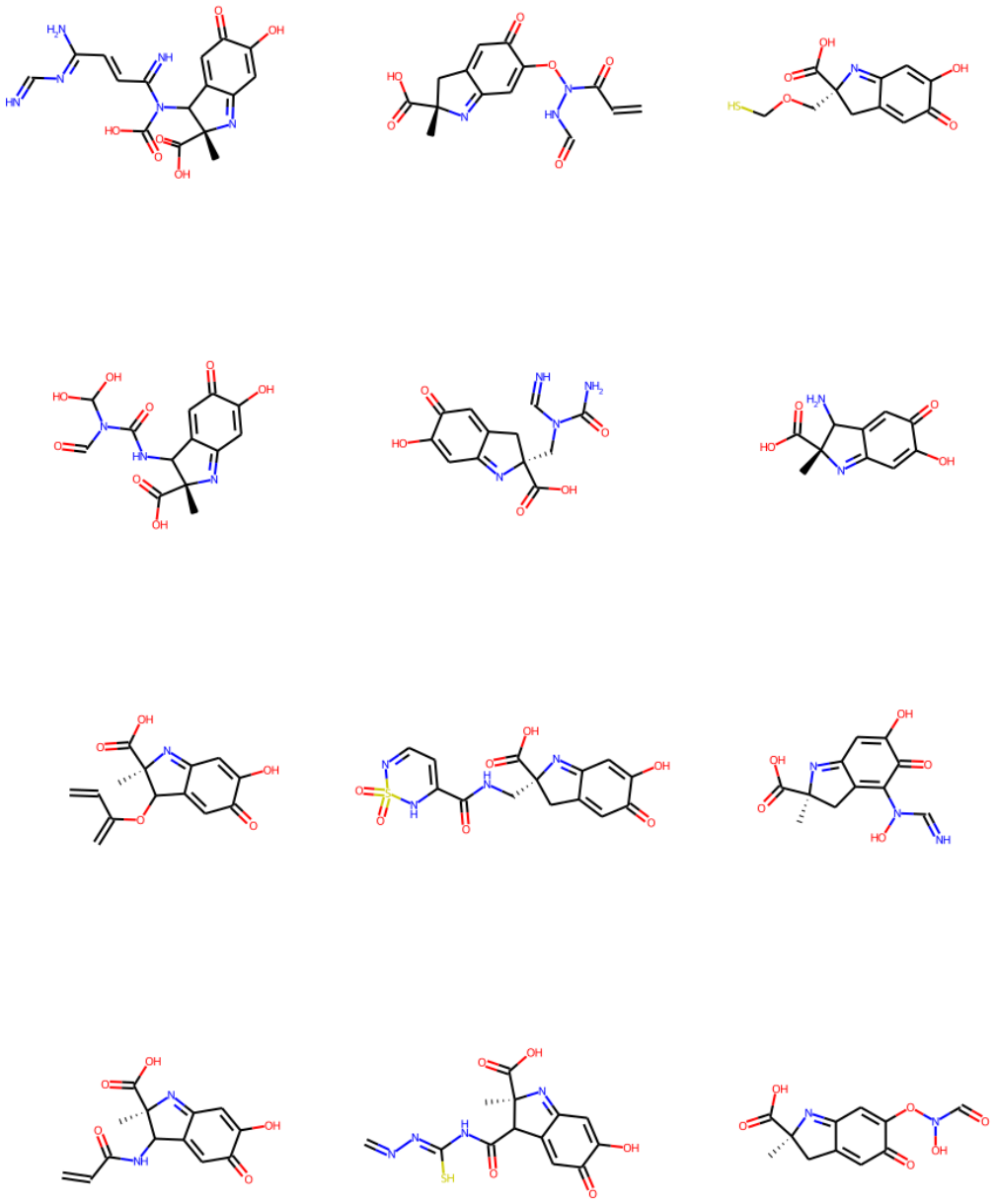


**Figure 4.23.** Molecules generated from the first molecule with perturbation based approach

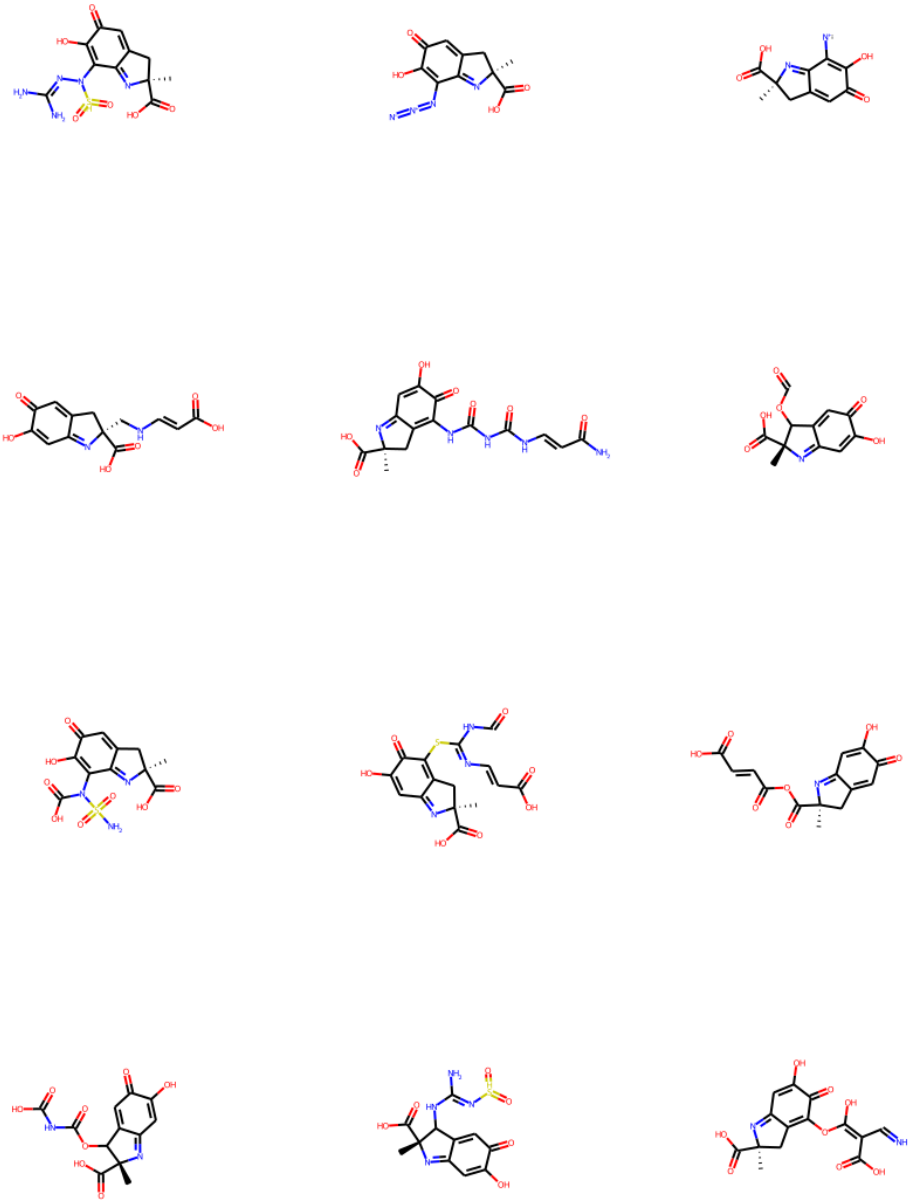


**Figure 4.24.** Molecules generated from the first molecule with IG

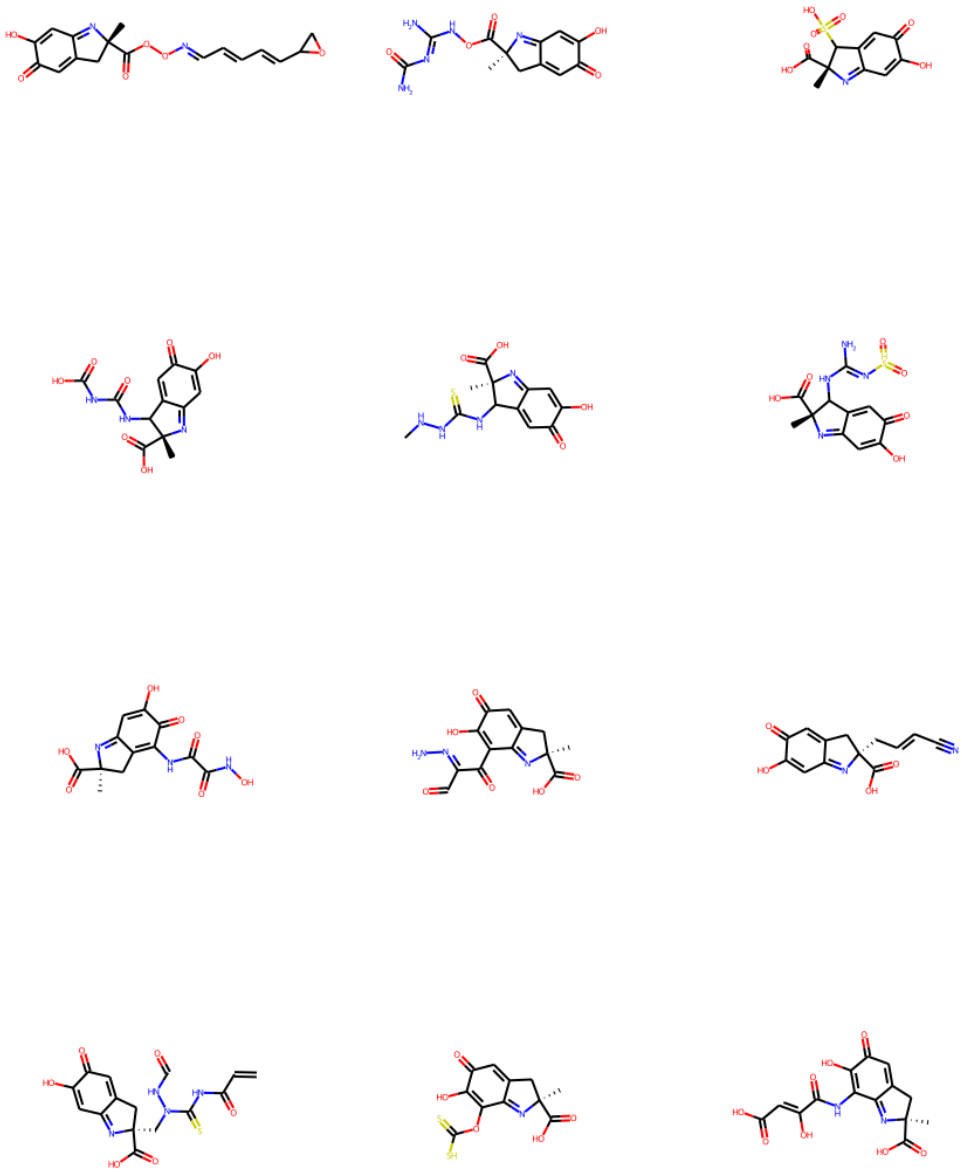


**Figure 4.25.** Molecules generated from the first molecule with DeepLIFT

| Method | Mean | Std | Min |
|---|---|---|---|
| Random | 5.48 | 3.35 | -4.38 |
| Perturbation | 4.93 | 3.48 | -4.26 |
| IG | 5.38 | 2.99 | -5.51 |
| DeepLIFT | 6.72 | 3.39 | -4.05 |

**Table 4.8.** Predicted pKa statistics of molecules generated from the first molecule

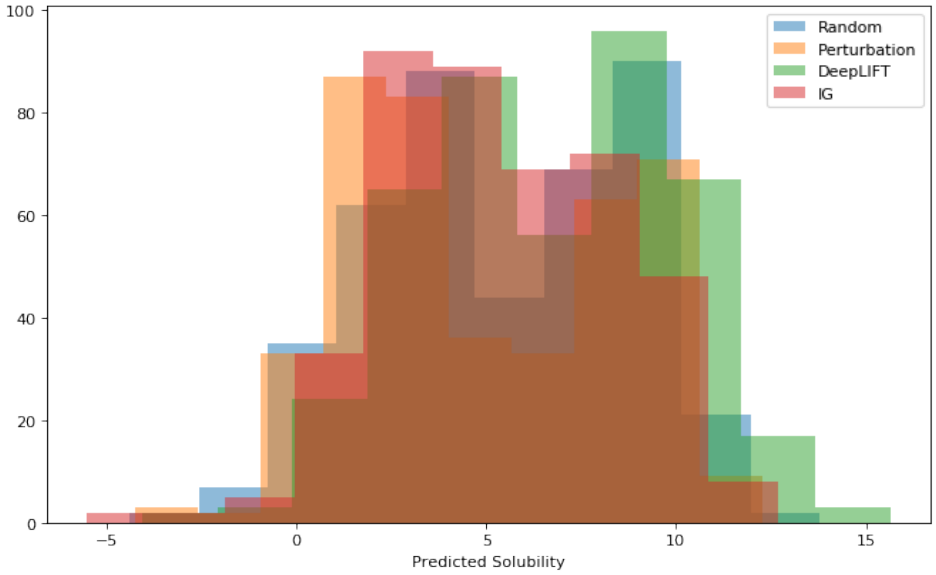



**Figure 4.26.** Predicted pKa distribution of molecules generated from the second molecule

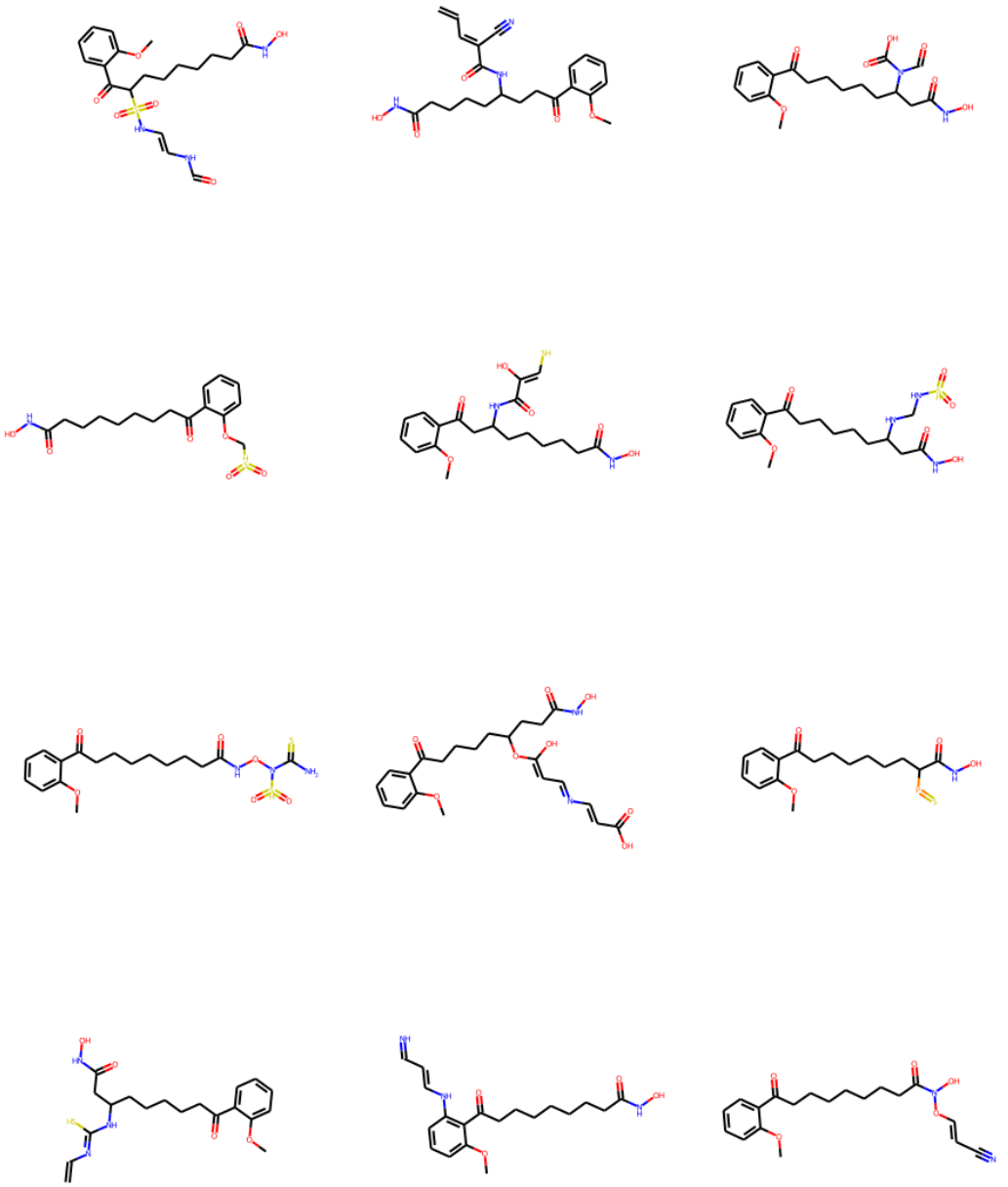


**Figure 4.27.** Molecules generated from the second molecule with random approach

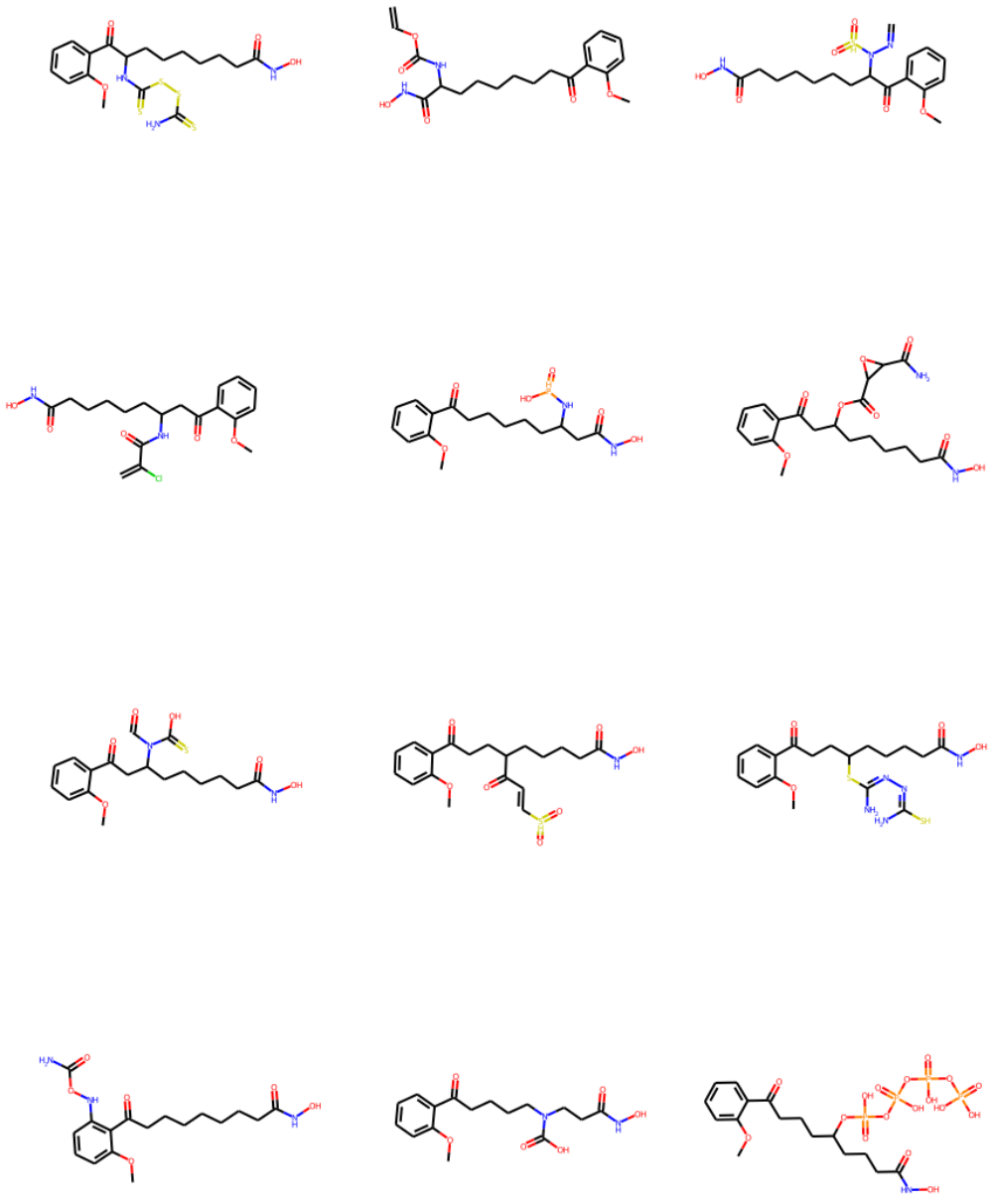


**Figure 4.28.** Molecules generated from the second molecule with perturbation based approach

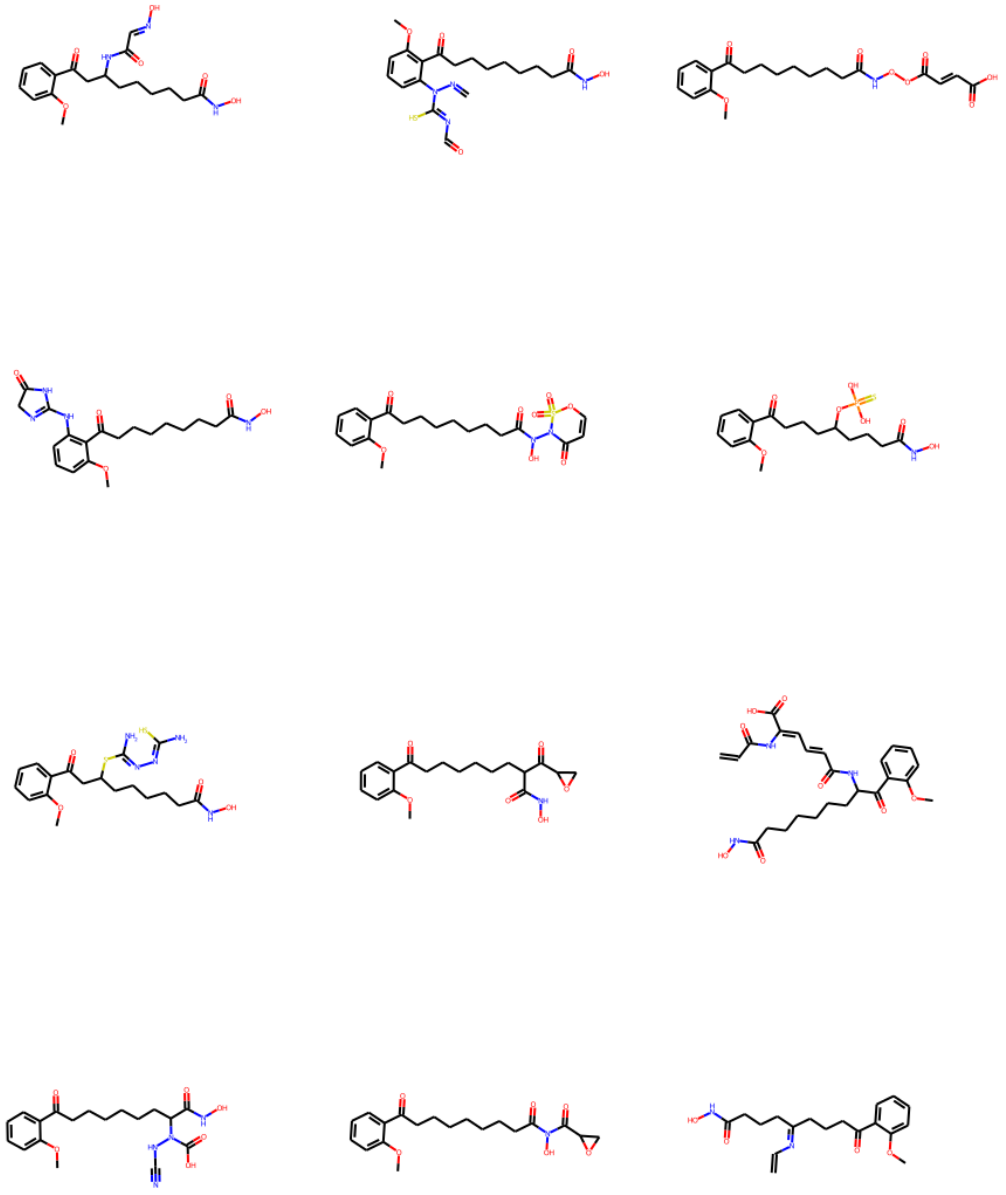


**Figure 4.29.** Molecules generated from the second molecule with IG

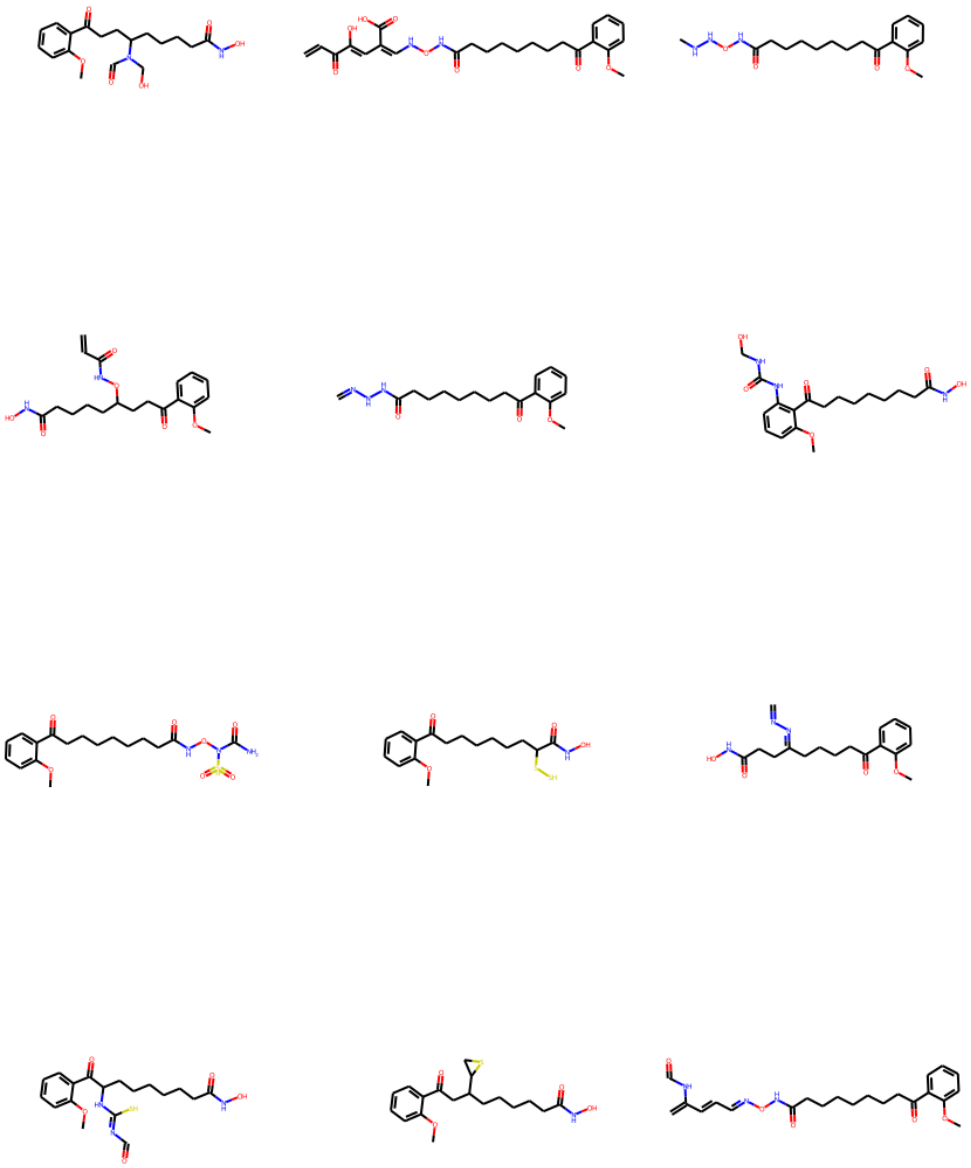


**Figure 4.30.** Molecules generated from the second molecule with DeepLIFT

# Chapter 5

# Conclusions

The continuous growth of NN parameters leads to new models with high performances but very difficult to interpret. This problem, also known as "Black Box" problem created the needs to develop new approaches to explain both the models and their predictions.

Using Neural Networks in drug design may help reducing the costs of lead discovery. Using GCN it is possible to exploit both chemical and structure information of the molecules.

Explainable Artificial Intelligence proved to be a key aspect for developing future models in drug design. In particular, using a perturbation based learning approach to train a GGN have shown it could help in studying the structure-activity relationships just using 2-D molecular representations.

Moreover, molecular attributions can be exploited to optimize molecules activities/properties adding or substituting atoms of portions of molecules using functional groups in a rational way.

A sister thesis by Savoia [2021] studies the application of saliency maps to the automatic generation of molecules aimed to the activity optimization. It is possible, in fact, to use different types of functional groups and to execute a targeted choice of the substituents basing on the attributions.

Future works might try to embed saliency maps in reinforcement learning agents in order to guide the model to generate molecules while taking into account other constraints such as the similarity to already existing drugs and the Med Chem Lipinski's rule of five (drug likeness). Moreover the usage of saliency maps for molecules generation might also help in studying the molecular space identifying particular species which are very similar but highly differ in activity, also known as activity cliffs.

# Acknowledgments

*I thank my advisor prof. Capobianco for following me during this study.*

*I am grateful to my father, who passed me down the love to this subject and guided me through my university path.*

*I also thank my mother, my sister Chiara and my girlfriend Veronica for being supportive during my whole academic career.*

*Last but not least, I am thankful to my friend Dylan with whom I shared my whole university experience.*

# Bibliography

Drug solubility challenge. URL https://www.kaggle.com/c/drug-solubility-challenge/overview.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10, 2015.

Richard Cramer, David Patterson, and Jeffrey Bunce. Comparative molecular field analysis (comfa). 1. effect of shape on binding of steroids to carrier proteins. *Journal of the American Chemical Society*, 110:5959–67, 08 1988. doi: 10.1021/ja00226a005.

John S. Delaney. ESOL: estimating aqueous solubility directly from molecular structure. *Journal of Chemical Information and Computer Sciences*, 44(3):1000–1005, May 2004. doi: 10.1021/ci034243x. URL https://doi.org/10.1021/ci034243x.

Peter Ertl. An algorithm to identify functional groups in organic molecules. *Journal of Cheminformatics*, 9(1):36, Jun 2017. ISSN 1758-2946. doi: 10.1186/s13321-017-0225-z. URL https://doi.org/10.1186/s13321-017-0225-z.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Spencer M. Free and James W. Wilson. A mathematical contribution to structure-activity studies. *Journal of Medicinal Chemistry*, 7(4):395–399, July 1964. doi: 10.1021/jm00334a001. URL https://doi.org/10.1021/jm00334a001.

Anna Gaulton, Louisa J. Bellis, A. Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(Database issue):D1100–D1107, Jan 2012. ISSN 1362-4962. doi: 10.1093/nar/gkr777. URL https://pubmed.ncbi.nlm.nih.gov/21948594. 21948594[pmid].

Giuseppe De Giacomo, Luca Iocchi, Marco Favorito, and Fabio Patrizi. Foundations for restraining bolts: Reinforcement learning with ltlf/ldlf restraining specifications. In *ICAPS*, 2019.

Corwin Hansch, Peyton Maloney, Toshio Fujita, and Robert Muir. Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients. *Nature*, 194:178–180, 04 1962. doi: 10.1038/194178b0.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL `http://arxiv.org/abs/1609.02907`.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch. *CoRR*, abs/2009.07896, 2020. URL `https://arxiv.org/abs/2009.07896`.

Greg Landrum. Rdkit: Open-source cheminformatics. URL `http://www.rdkit.org`.

Michael Lent, William Fisher, and Michael Mancuso. An explainable artificial intelligence system for small-unit tactical behavior. pages 900–907, 01 2004.

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de-novo drug design. *CoRR*, abs/1711.10907, 2017. URL `http://arxiv.org/abs/1711.10907`.

Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. Molecularrnn: Generating realistic molecular graphs with optimized properties. *CoRR*, abs/1905.13372, 2019. URL `http://arxiv.org/abs/1905.13372`.

Rino Ragno. www.3d-qsar.com: a web portal that brings 3-d qsar to all electronic devices—the py-comfa web application as tool to build models from pre-aligned datasets. *Journal of Computer-Aided Molecular Design*, 33(9):855–864, Sep 2019. ISSN 1573-4951. doi: 10.1007/s10822-019-00231-x. URL `https://doi.org/10.1007/s10822-019-00231-x`.

Arun Rai. Explainable ai: from black box to glass box. *Journal of the Academy of Marketing Science*, 48(1):137–141, Jan 2020. ISSN 1552-7824. doi: 10.1007/s11747-019-00710-5. URL `https://doi.org/10.1007/s11747-019-00710-5`.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, April 2010. doi: 10.1021/ci100050t. URL `https://doi.org/10.1021/ci100050t`.

Dylan Savoia. Molecule generation from input-attributions over graph convolutional networks. 2021.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128 (2):336–359, Oct 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL `http://dx.doi.org/10.1007/s11263-019-01228-7`.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

Alex Zhavoronkov, Yan A. Ivanenkov, Alex Aliper, Mark S. Veselov, Vladimir A. Aladinskiy, Anastasiya V. Aladinskaya, Victor A. Terentiev, Daniil A. Polykovskiy, Maksim D. Kuznetsov, Arip Asadulaev, Yury Volkov, Artem Zholus, Rim R. Shayakhmetov, Alexander Zhebrak, Lidiya I. Minaeva, Bogdan A. Zagribelnyy, Lennart H. Lee, Richard Soll, David Madge, Li Xing, Tao Guo, and Alán Aspuru-Guzik. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature Biotechnology*, 37(9):1038–1040, September 2019. doi: 10.1038/s41587-019-0224-x. URL `https://doi.org/10.1038/s41587-019-0224-x`.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015.