Do machines dream of atoms? Crippen's logP as a quantitative molecular benchmark for explainable AI heatmaps

Maria H. Rasmussen[1,2], Diana S. Christensen[1], and Jan H. Jensen[1,2]

[1]Department of Chemistry, University of Copenhagen, Denmark
[2]E-mail: mhr@chem.ku.dk, jhjensen@chem.ku.dk, Twitter: @janhjensen

December 1, 2022

Abstract

While there is a great deal of interest in methods aimed at explaining machine learning predictions of chemical properties, it is difficult to quantitatively benchmark such methods, especially for regression tasks. We show that the Crippen logP model (*J. Chem. Inf. Comput. Sci.* 1999, 39, 868) provides an excellent benchmark for atomic attribution/heatmap approaches, especially if the ground truth heatmaps can be adjusted to reflect the molecular representation. The "atom attribution from finger prints"-method developed by Riniker and Landrum (*J. Chem. Inf. Comput. Sci.* 2013, 5, 43) gives atomic attribution heatmaps that are in reasonable agreement with the atomic contribution heatmaps of the Crippen logP model for most molecules, with average heatmap overlaps of up to 0.54. The agreement is increased significantly (to 0.75) when the atomic contributions are adjusted to match the fact that the molecular representation is fragment-based rather than atom-based (the finger print-adapted (FPA) ground truth vector). Most heatmaps and the corresponding FPA overlaps are relatively insensitive to the training set size and the results are close to converged for a training set size of 1000 molecules, although for molecules with low overlap some heatmaps change significantly. Using the "remove an atom" approach for graph convolutional neural networks (GCNNs) suggested by Matveieva and Polishchuk (*J. Cheminform.* 2021, 13, 41) we find an average heatmap overlap of 0.47 for the atomic contribution heatmaps of the Crippen logP model. Like the simpler attribution benchmarks for classification tasks that have come before it, this work sets the bar for regression tasks.

1 Introduction

Machine learning (ML) models occasionally make wrong predictions and given their black-box nature it is not always obvious when that is the case. While there are several methods for assigning uncertainties to the predictions, these methods report (at best) on the likelihood of prediction errors and there is not a strong correlation between errors in the predictions and their uncertainties.[1] There is therefore a great deal of interest in methods aimed at explaining the ML predictions, often referred to as explainable AI (XAI), which can help humans decide whether the predictions are reasonable.[2, 8, 9, 10, 11, 12, 13, 14, 15, 3, 4, 5, 6, 7] Within chemistry, in general, and drug discovery in particular, another motivation is to use the explanation to help guide the design of molecules with improved properties (for example: [16]).

Attribution methods, which aim at producing explanations by assigning a numerical value to each atom to create a so-called heatmap, are among the most popular XAI methods in chemistry.[13, 14, 15, 3, 4, 5, 6, 17, 7] Some of these methods were originally developed for image classification (for example [18]) where it is often fairly obvious whether the heatmap highlights the correct part of the image. However, for chemical applications it is often less clear whether the atomic attributions are correct for a certain chemical property, which complicates the benchmarking of these methods. One solution is to use simple toy models such as classifying molecules with respect to the presence or absence of certain functional group. While methods that fail at such simple tasks can probably be discounted, it is not clear whether methods that succeed will also succeed for more complex classification and, especially, regression tasks.

Harren at al.[13] has demonstrated that binding affinity data for pairs of closely related molecules (matched molecular pairs) combined with expert chemical knowledge can be a powerful benchmark, but it is difficult to quantify the performance using this approach. Sanchez-Lengeling et al.[5] addressed this problem by fitting models to experimentally measured solvation energies[19] and comparing the corresponding heatmaps to the contributions from Crippen's well-known linearly additive atom-based model of logP values[20] - a property that is related to solubility. The attribution methods tested using this benchmark gave atomic attributions with relatively modest correlation to this ground truth but, as pointed out by Henderson et al.[14], part of the reason may be that the correlation between logP values and solvation energies is not perfect. Instead they suggested that it may be better to fit the model to Crippen logP values themselves, but did not test this approach.

In this study we show that ML models fit to Crippen logP values do lead to heatmaps that are in slightly better agreement with the ground truth heatmap derived from the atomic contributions of the Crippen model. However, when using finger prints (FPs) as the molecular representations there is a fundamental limit to the correlation that can be obtained due to the fact that the FPs are inherently fragment based and not atom based. We show that when the ground truth heatmaps are adapted to reflect this fragment-based nature the correlation is increased significantly.

While graph-based approaches accept input that is more atom based than FPs, we generally find worse agreement with the ground truth for the GCNN/XAI combi studied here. Through these examples we exemplify how a quantitative regression benchmark such as the one presented can be used to extract information about the behavior of the ML model and/or XAI method.

2 Computational Methodology

The Crippen logP model[20] implemented in RDKit[21] predicts the logP value of a molecule by

$$\text{logP} = \sum_i n_i a_i \tag{1}$$

where n_i is the number of a particular atom type i and a_i is the logP contribution from that atom type. There are about 100 different atom types and their contributions were determined by fitting to experimental logP values. The atom types are defined by the nearest neighbour atoms so that, for example, the a_i for C is different for ethane and methanol. However, the model is local in the sense that a_i is independent of atoms not directly bonded to atom i. The a_i values are taken as the ground truth for atom attributions, with the caveat that a_i values from H atoms are added to the closest non-H atom.

We use the Random Forest (RF) regression model implemented in scikit-learn,[22] with 200 trees and a minimum of three samples per leaf node. The predicted logP value is the average over all trees and the uncertainty in the prediction is the corresponding standard deviation. The molecules for the training and test set are taken from a 250k molecules subset of the ZINC data base, which has been used in many other studies.[23, 24, 25, 26] A training set of size N corresponds to the first N molecules in the data set and the 5k test set corresponds to the last 5k molecules in the data set. For the molecular representation we use Morgan extended connectivity fingerprints[27] with a diameter of four (ECFP4) as implemented in RDKit. This method identifies fragments of varying sizes centered at each atom of the molecule, with the maximum size determined by the radius. Each fragment is then assigned a random position in a binary vector of length 2048, where the presence and absence of a particular fragment is indicated by a 1 and 0, respectively. Other bit-vector sizes are also possible, but 2048 is a very typical value. The number of different fragments for a

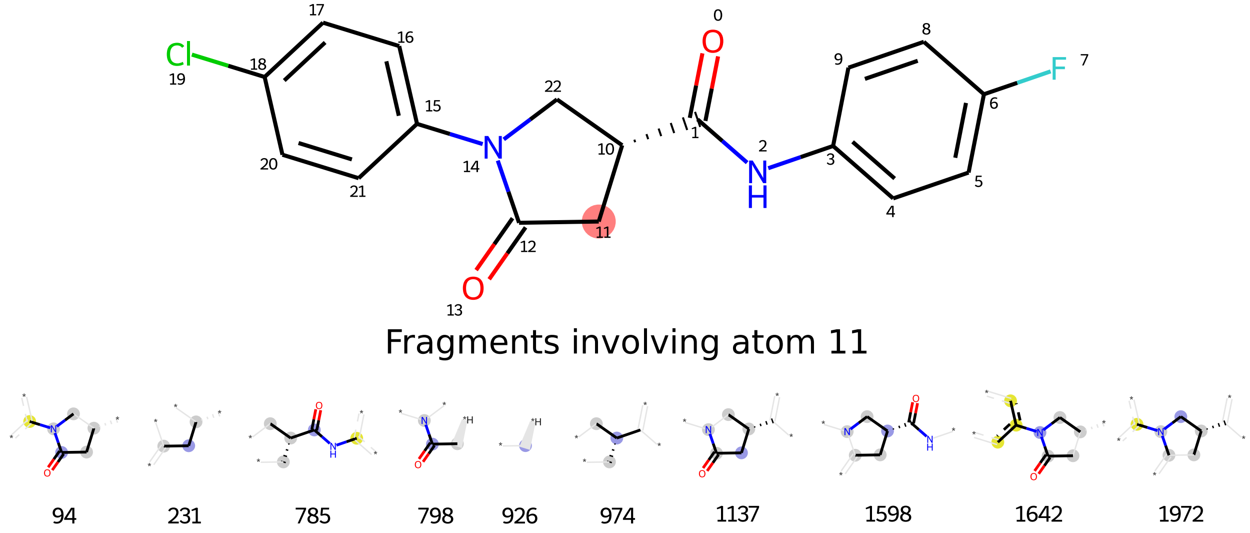


Figure 1: An example of the fingerprint fragments that are removed when an atom (11) is removed. Blue circle: central atom, yellow circle: aromatic atom, grey circle: aliphatic ring atom, star/light gray bond: atom/bond not directly part of the fragment but affecting connectivity.

collection of molecules is typically much larger than 2048, meaning that a bit position can report on many different fragments, which is known as bit collision.

We use the “atom attribution from finger prints”-method developed by Riniker and Landrum[3] and also test the dummy-atom approach as implemented by Jimenez-Luna et al.[15] Both methods are described in more details in the following section. The atom attributions are visualised as colored contour plots, with magenta-colored dotted and green-colored solid contour lines for negative and positive contributions, respectively. When drawing the maps of the atomic contributions the number of contour lines from minimum to maximum value needs to be set. This is set so that each contour line approximately represents a 0.06 change: $N_{contour} = \frac{ac_{max}-ac_{min}}{0.06}$, where ac_{max} and ac_{min} are the highest and lowest atomic contributions of the molecule, respectively. $N_{contour}$ is rounded to nearest integer. The coloring is scaled to the maximum absolute value in the attribution vector.

3 Results and Discussion

3.1 Comparison of the atomic attributions to the ground truth

The “atom attribution from finger prints”-method developed by Riniker and Landrum[3] computes the contribution of a given (non-hydrogen) atom by removing all bits from the fingerprint for which the corresponding fragments contain the atom (cf Figure 1). The ML predicted value using this new fingerprint is subtracted from the value predicted with the unmodified fingerprint and the difference is attributed to that atom. This results in a vector of atom attributions that we want to compare to the corresponding atom contributions-vector (the ground truth vector) from the Crippen logP model. While the ground truth vector sums up to the ground truth logP value, the atom attribution vector does not sum up to the ML-predicted logP value. The vector elements are in fact very different in magnitude so a simple difference is not instructive. Sanchez-Lengeling et al.[5] used Kendall’s tau (rank correlation) while Henderson et al.[14] used Pearson’s r to quantify agreement. We choose to compute the dot product of the normalised atom attribution and ground truth vectors, which ranges from -1 to 1 and where 1 corresponds to a perfect agreement. This overlap compares the distribution and relative importance of positive and negative contributions within the molecule, but it does not report directly on the contribution of each atom to the logP value. Thus, when comparing the vectors visually we re-scale the attribution vector so that it sums to the predicted logP value and depict the magnitude of these contributions as a contour map, while the color intensity corresponds to a "normalised" vector where the largest magnitude contribution is 1 (this vector is very similar to the normalised vectors used to compute the overlap, but gives better visual comparison). We compare the overlap to Pearson's r below.

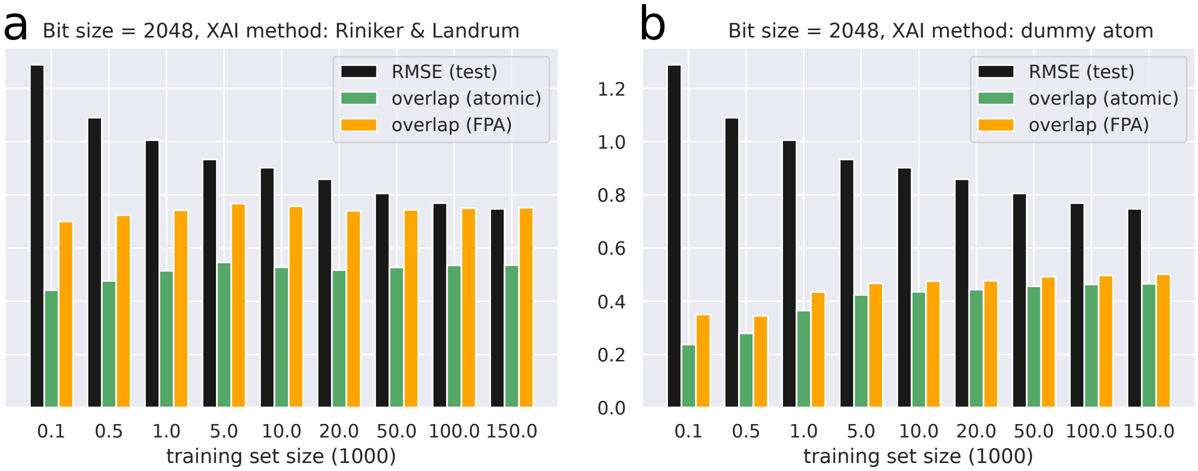


Figure 2: (a) The average overlap of the ML vector with different versions of the ground truth and FPA vectors calculated based on the RDKit atom contributions. (b) Same as for (a) but using the dummy atom approach. Note that the ML models and therefore model error (RMSE bars) used in (a) and (b) are the same.

3.2 Results for a large training set

We train nine different RF/ECFP4 models using training sets ranging in size from 100 to 150k molecules and test the performance using a test set of 5k molecules. The RMSE is shown in Figure 2a and b and suggests that the error is converged for the largest training set and that the model is as good as it is going to get. We first focus on the results from this model since that separates any issues related to incomplete training from any issues intrinsic to the XAI methodology.

The average overlap (green column) is 0.54 indicating that the atomic attributions are largely correct for a majority of the molecules. For comparison, a null-model attribution vector, where each atom is assigned the value logP/N where logP is the predicted value and N is the number of non-H atoms, results in an average overlap of 0.32 (Figure S2). Figure 3a and b shows plots of the atomic attributions and the ground truth contributions, respectively, for a molecule (**1**) with an overlap (0.66) close to the average. Both plots show positive contributions from the phenyl rings, but the rest of the atoms in the molecule do not appear to contribute significantly to the predicted logP value, in contrast to the ground truth. Never-the-less the logP value is predicted to within 0.3 units. One possible reason is that when using a FP radius of two, the removal of an atom results in the removal of a relatively large chunk of the molecule, while the associated change in logP is ascribed to a single atom. For example, removing the carbonyl C atom in the pyrrolidone ring actually removes the entire moiety plus part of the substituents (Figure 1) and the combined atomic logP contributions of these atoms, which is roughly zero, is assigned to that atom. To quantify this effect we compute the sum of the logP contributions for each of the ten fragments and assign it to the carbonyl C and

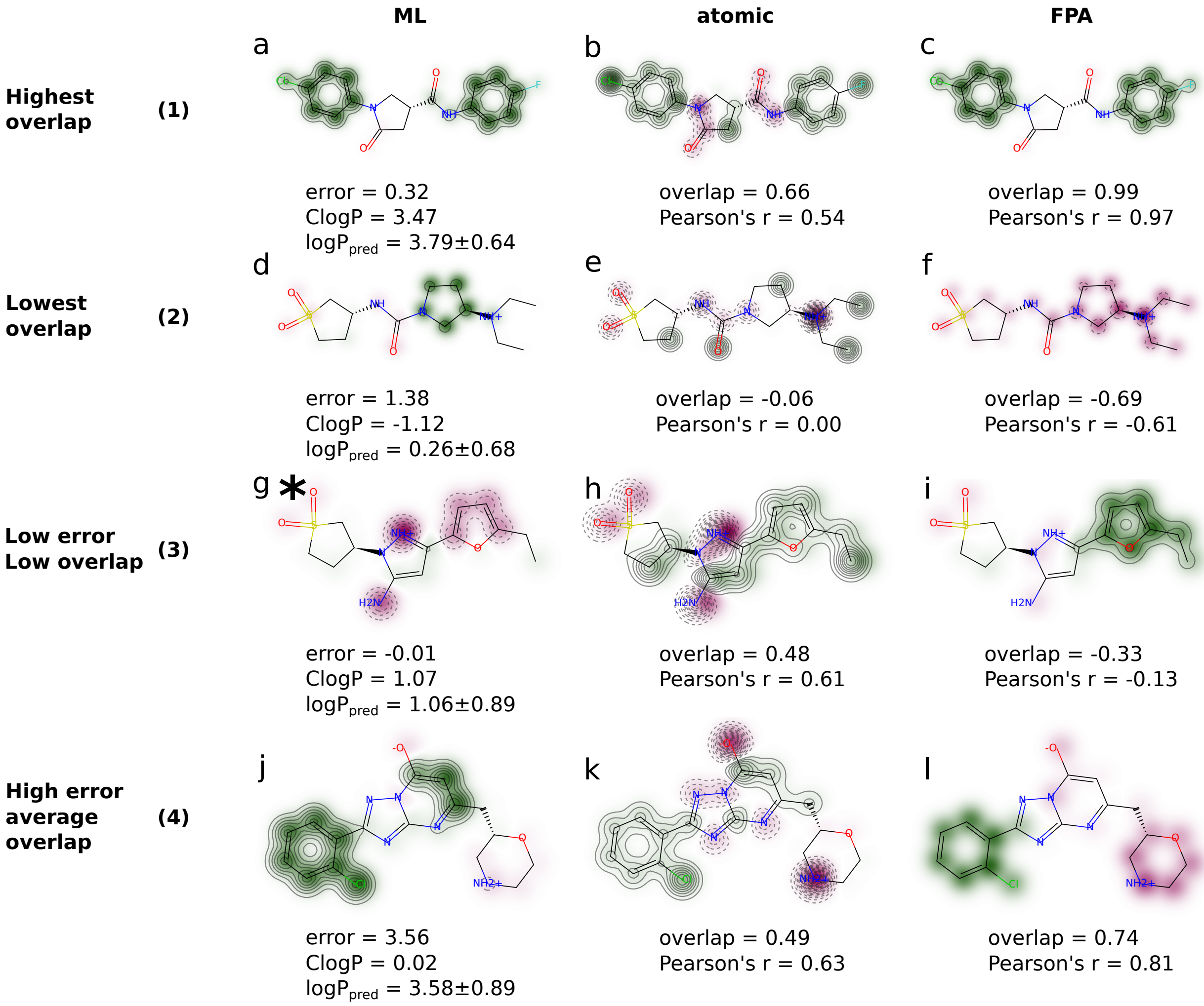


Figure 3: Heatmap examples for four molecules (**1-4**). The first column is the heatmap from the ML model trained on 150k molecules. The second column is the ground truth heatmap from the atomic contributions of the Crippen model and the third column is the FP adapted ground truth heatmap (see text). ML contributions are scaled so they sum to ML prediction, AF contributions are scaled to sum logP. Crippen (atomic) contributions sum to logP by nature. *While the predicted logP is positive, summing the atomic contributions results in a negative number. Instead of scaling to logP$_{\text{pred}}$, the atomic contributions are scaled to sum to -logP$_{\text{pred}}$

repeat this process for all the other atoms to produce a “finger print-adapted” (FPA) ground truth vector. A plot of this vector is shown in Figure 3c and shows a near perfect agreement with the ML attribution vector, with an overlap of 0.99. Thus, the discrepancy between the attribution and ground truth vectors observed for this molecule is due to the way the attributions are computed and not a deficiency in the ML model itself. Using FPA ground truth vectors the average overlap for the test set increases significantly from 0.54 to 0.75, suggesting that this is the case for most molecules, although the average null-model overlap also increases, to 0.61 (Figure S2).

However, the good match between the ML-attribution and FPA vectors for **1** does not mean that the ML model has learned the contribution of each FP fragment correctly. Out of the 41 different ECFP4 fragments that describe **1** only nine (Figure 4) make a significant (>0.05) contribution to the logP value and a FP with only these nine bits reproduces the predicted logP value to within 0.25 units. All but one of these fragments (fragment 90) are ECFP2 fragments and none of these fragments derive from the pyrrolidone ring. So according to the ML model the net logP contribution of the pyrrolidone ring is nearly zero because all associated fragments each make nearly zero contributions, in contrast to the FPA vector where most fragments make sizeable positive or negative contributions that mostly cancel.

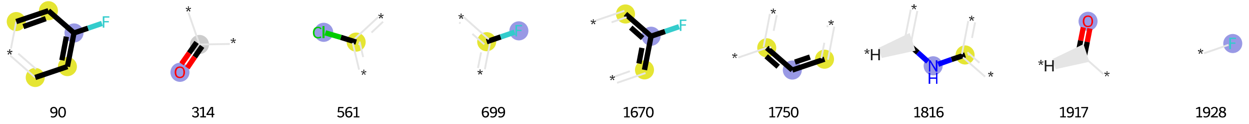


Figure 4: The bits that make the largest contributions to the predicted logP value of molecule **1**. Blue circle: central atom, yellow circle: aromatic atom, grey circle: aliphatic ring atom, star/light gray bond: atom/bond not directly part of the fragment but affecting connectivity.

Figures 3d-f show similar plots for the molecule (**2**) with the lowest FPA overlap (-0.69) in the test set. The predicted logP value is slightly positive (0.26) while the ground truth is negative (-1.12) and the error (1.38) is more than twice the model MAE (0.58) (Table S1). Comparison of the ML-attribution vector to the ground truth and FPA vectors clearly show that the discrepancy arises from the region of the molecule involving the N cation. Indeed, the error is eliminated by removing the proton, which also increases the overlap to 0.75 (Figure S3).

In general a low overlap does not necessarily correspond to a high error: the Pearson correlation factors for the FPA overlap vs MAE is only -0.11. While the direct correlation is small there is a modest enrichment of low error predictions for molecules with high overlap (Figure S7d). For example, the MAE of molecules with overlaps between 0.8 and 1.0 is 0.55 while the MAE for molecules with negative overlaps is 0.82

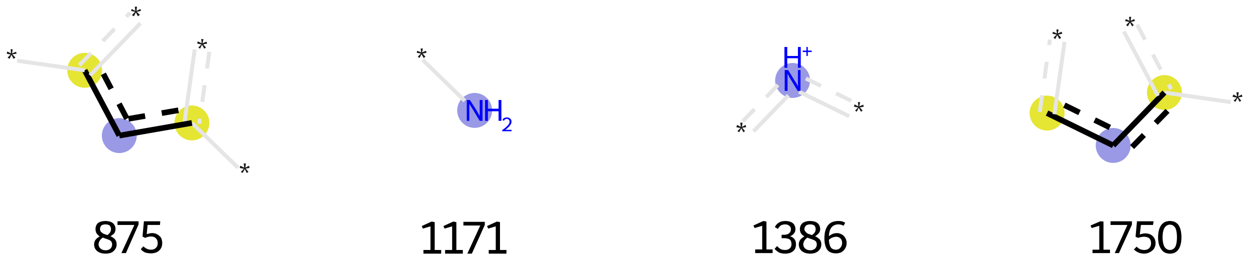


Figure 5: The bits that make the largest contributions to the predicted logP value of molecule **3**. Blue circle: central atom, yellow circle: aromatic atom, grey circle: aliphatic ring atom, star/light gray bond: atom/bond not directly part of the fragment but affecting connectivity.

Figures 3g-i show heat maps for a molecule (**3**) with both a low error and low FPA overlap (-0.33). In this case the ML-attributions are all negative, while the predicted logP value is positive. Comparison of the ML attribution vector and FPA vector shows that the problem lies mainly with the C atoms in the furan ring, which make negative contributions to the logP. Removing the N proton changes the sign of these contributions and increases the overlap to 0.73 (Figure S4), so the "sign problem" in the furan ring is related to the protonation state of the neighboring pyrazole ring. We investigated several possible explanations, such as bit collisions between fragments in these two rings or the effect of FP fragments that connect the two rings, but the reason turns out to be a bit more complicated. The changes in predicted logP that gives rise to the heat map has three main contributions (Figure 5): bit 1171 (the NH_2 group), 1386 (the NH^+ group), and 1750 (the furan C atoms). In fact a FP vector with only these three bits results in a predicted logP value (0.91) that is very similar to the value predicted with all 45 on-bits (1.07). Removing bit 1750 from this FP decreases the predicted logP by 0.36 (i.e. bit 1750 makes a positive logP contribution), while removing bit 1750 from the full FP vector increases logP by 0.35, which gives rise to the negative contours on the furan ring. Clearly, at least one additional bit is needed for the negative furan ring contributions and that bit turns out to be 875, which represents the three C atoms of the pyrazole ring (Figure 5). Note that all three bits (875, 1171, and 1386) must be on for bit 1750 to make a negative contribution, which is why deprotonation (which removes bit 1386) changes the sign of bit 1750's contribution. This is a case of overfitting in the sense that the ML model has learned a non-additive rule for an additive property, but we will qualify this point further at the end of this subsection.

Clearly, the ML heatmap for **3** (Figure 3g) is not helpful in understanding the predicted logP values and we found that for 13% of the molecules the sum of the ML atomic attributions do not have the correct sign (Figure S7d). This sign problem is also found in the FPA heatmaps in 5% of the molecules, but only 2% of the molecules have a sign problem for both the ML and FPA heatmaps. About half of the sign problems in the FPA heatmaps occur for logP values that are ≤ 0.5 suggesting the sign problem is due to imperfect cancellation of nearly equal positive and negative contributions and the heatmaps still offer insights into the predicted logP value (see e.g. Figure S4).

The non-additive behavior observed for compound **3** is also observed for compound **2**: at first sight the heat map seems to show that the model simply has erroneously learned that a protonated tertiary amine group makes a small contribution to a logP value. However, a similar bit-analysis shows that the NH^+ only makes small contributions when bits related to the distant sulfone group are on. Changing the sulfone group to a methylene group leads to an excellent logP prediction (0.53 vs 0.64) and a better overlap (0.60) where the NH^+ makes a sizable negative contribution to the logP value (Figure S3).

Figures 3j-l show heat maps for the case of high error but average overlap. The relatively high overlap considering the difference in heatmaps reflect the fact that the overlap focuses more on the distribution of positive and negative contributions, rather than on their magnitudes. The ground truth and FPA vectors show that the ground truth logP value (0.02) is a result of the near perfect cancellation of positive and negative contributions of near equal magnitude. Comparing the ML- and FPA-vectors it is clear that while the colors, which are representative of the overlap, are matched reasonably well, the contours, which are representative of the magnitude of the contributions, are not. The error clearly comes from an underestimation of the effect of the N cation and an overestimation of some of the C atoms in the pyrimidine ring. Neutralising the cationic N decreases the error by ca 1 unit due the ground truth logP value being raised by roughly the same amount while the predicted value is essentially unchanged (Figure S5). However, roughly the same error (2.26) can be obtained by removing the chlorophenyl group since the NH_2^+ group now correctly makes a sizeable negative contribution to the predicted logP value (Figure S5). So, just like for **2**, the contribution of a cationic N is dependent on another, distant, functional group, which is clearly at odds with the additive nature of the ground truth model. In the absence of the proton or the chlorophenyl group, the remaining error is largely due to some of the C atoms in the pyrimidine ring. The reason seems to be a consistent overestimation of the effect of the corresponding bit (875, Figure 5) for the pyridine ring and the overestimation can be removed simply by adding a methyl group (Figure S5).

Sheridan[4] has developed an XAI approach in which the contribution of an atom is determined by replacing it with a Na atom and this approach has been adopted by others.[15, 13] The advantage if this approach is its ease of implementation and general applicability to all kinds of molecular representations such as FPs and graph neural networks. Figure 2b shows the average overlaps as a function of training set size using this approach as implemented by Jimenez-Luna et al.[15] The overlaps are significantly lower, especially using the FPA vector, where the average overlap is lower than the null-model for all training set sizes. The effects on the average overlaps with the ground truth vector is less pronounced but the average overlaps are similar to or lower than the null-model for training set sizes smaller than 5000 molecules.

The likely reason for the poorer performance of the dummy atom approach is that while it removes the same bits as the Riniker and Landrum approach, it also introduces an equal number of new on-bits associated with the dummy atom. These new on-bits introduces spurious logP contributions which can corrupt the heatmaps. We note that these conclusions are specific to the FP representation and do not necessarily apply to the use of dummy atoms with graph neural networks.

To sum up, the "atom attribution from finger prints"-method developed by Riniker and Landrum, when used with a RF model, gives atomic attributions that are in reasonable agreement with the atomic contributions of the Crippen logP model for most molecules. The agreement is increased significantly when the atomic contributions are adjusted to match the fact that the molecular representation is fragment-based rather than atom-based (the FPA ground truth vector). Molecules where the atomic attribution differs significantly from the ground truth tend to have slightly larger errors on average, but the correlation factor is near zero when considering individual molecules. One question is whether cases such as **3** with low errors and wrong attribution is due to overfitting. The fact that the ML-model has learned non-additive correlations between structurally distant FP fragments is at odds with the additive and local nature of the ground truth and thus consistent with overfitting. On the other hand, the problem (as measured by the percentage of molecules with "sign problems") is most severe for the largest training set where one would expect overfitting to be less important. A likely explanation is that the use of a non-additive representation (binary FPs) to model an additive property, combined with bit collision, results in an intrinsically overfit model. For example, only two FP fragments make significant contributions to the predicted logP of n-butane: CH3- and CH2-CH3-CH2-. The fragment that indicates that there are two of each of these fragments in n-butane (CH3-CH2-CH2-CH3) is not in the training set so that bit position is used for another fragment not contained in n-butane (bit collision). Even if n-butane was contained in the training set, the corresponding fragment would only appear once and the model would learn that this bit position is far more likely to report on some other non-pertinent fragment. For example, there are only 24 instances where bit 94 corresponds to the fragment shown in Figure 1, compared to 6050 instances for the most popular fragment, so this bit cannot be used by the model to predict logP for molecule **1**. Thus, rather than making use of large fragments to help estimate the number and proximity of smaller fragments, the model is forced to learn correlations between smaller fragments. As a result, spurious correlations between fragments such as those seen for molecules **2**-**4** can occur because the representation does not effectively report on whether, for example, the fragments shown in Figure 5 are in the same ring or not. A corollary of this hypothesis is that the Crippen logP values represent a challenging benchmark for the ECFP4/RF model and, hence, the atomic attribution method.

As we have shown above, one can learn a great deal about the ML model from heatmaps by comparing to the corresponding ground truth heatmaps. However, ground truth heatmaps will not be available for other properties so tools for assessing the quality of the heatmap in the absence of the ground truth would be very useful. We investigate two approaches: uncertainty in the atomic attribution (UAA) and atomic attribution of the uncertainty (AAU). The UAA is obtained by computing the atomic attributions for each tree in the random forest and computing a standard deviation for each atom while the AAU is computed as described above for logP except using the standard deviation of the logP predictions instead of the logP value itself. A more detailed description as well as examples for the four molecules described above can be found in Section S1. We find that while these uncertainty-guided heatmaps can to some extent be used to indicate problematic regions of the molecules, the predictive power of either of the approaches is not strong enough to make them especially useful for this model/XAI combination.

3.3 Results for smaller training sets

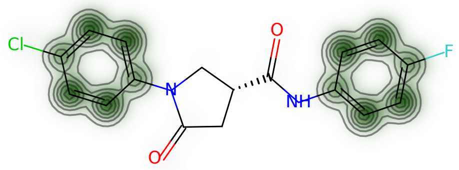
Most data sets in chemical science are often considerably smaller than 150k with many as small as 500-5000 molecules, so we investigate the effect on training set size on the conclusions drawn thus far. Figure 2a shows that the average overlaps are essentially converged for 5k molecules. The drop in average overlap on going to smaller training sets is more pronounced for the ground truth vector compared to the FPA vector but even for a training set of 100 molecules the average overlaps (0.44 and 0.70) are still significantly larger than for the null-model (0.31 and 0.58). Figures 6 and 7 show heat maps for molecules **1-4** for training set sizes of 100 and 1000 molecules, respectively. Comparison to Figure 3 shows that for molecule **1** there is very little change in the heatmap, overlaps, and predicted logP value on going to the smaller training set. For molecule **2** the heatmap for 100, 1000, and 150k all look different from the ground truth and from each other while the error in the predicted logP is consistently high. In contrast, for molecule **3** the FPA overlap decreases from 0.77 to 0.70 to 0.33 on going from training set sizes of 100 to 1000 to 150K, while the error in the predicted logP value decreases from 1.91 to 0.58 to -0.01. The furan ring is correctly predicted to make a positive contribution to the logP for the two smaller training set sizes and the heatmaps are consistent with a positive logP value. More generally, the number of molecules with "sign problems" is lower for the smaller training set sizes as shown in Figure S7. Finally, for molecule **4** the main difference in the heatmaps is the growing contribution of some of the C atoms in the pyrimidine ring on going to larger training set sizes, but the FPA overlap is essentially unchanged.

To sum up, most heatmaps and the corresponding FPA overlaps are relatively insensitive to the training set size and the results are close to converged for a training set size of 1000 molecules, although for molecules with low overlap some heatmaps change significantly. The difference in average error for molecules with high and low overlap is more pronounced for smaller training sets (Figure S7), but this could just reflect the larger spread in errors for models trained on smaller training sets.

3.4 Overlap vs correlation coefficients

As mentioned in the introduction, Sanchez-Lengeling et al.[5] have tested graph-based atom attribution methods by fitting solubility data[19] and then comparing the atomic attributions to the atomic contributions from the Crippen logP model - an approach that has also been used by Henderson et al.[14]. Both used correlation coefficients to compare the vectors for individual molecules: Sanchez-Lengeling et al. used Kendall's tau (rank correlation) while Henderson et al. used Pearson's r. In general we find that there is a good correlation between Pearson's r and the overlap. For example, for the 150k training set the average Pearson's r values for the ground truth and FPA ground truth are 0.46 and 0.74, which are in reasonable agreement with the corresponding average overlaps of 0.54 and 0.75 (Table S2). However, comparing R and overlap values for the four molecules in Figure 3 the R value occasionally overestimates the agreement with the ground truth vector. For example, for molecules **3** and **4** the R values indicate significantly above-average agreements with the ground truth vector, while the overlaps indicate slightly below-average agreements.

Since the models used by Sanchez-Lengeling et al. and Henderson et al. are fit to solubility values rather than logP values one cannot say anything definitive about how their attribution methods compares to our approach. However, the best average r values found by Sanchez-Lengeling et al. and Henderson et al. (0.37 and 0.28) are not too different from the corresponding value (0.43) we obtain for a training set size of 1000 (Table S2), which roughly corresponds to the size of the solubility dataset used in these studies. As pointed out in both studies, these average r values are relatively low and our study suggests that perhaps they could be improved somewhat by fitting the model to the Crippen logP values. However, a more important



1

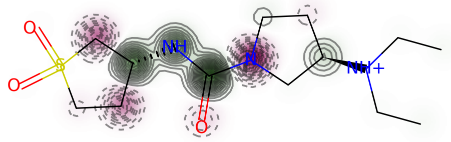
error = -0.42

ClogP = 3.47

$\mathrm{logP}_{\mathrm{pred}}$ = 3.05±0.99

overlap (ground truth) = 0.63

overlap (FPA) = 0.97



2

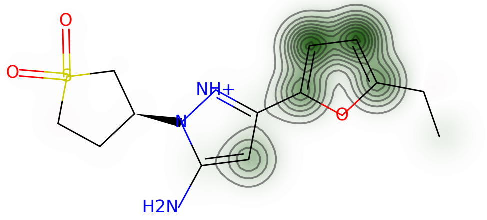
error = 1.53

ClogP = -1.12

$\mathrm{logP}_{\mathrm{pred}}$ = 0.41±1.44

overlap (ground truth) = 0.15

overlap (FPA) = 0.05



3

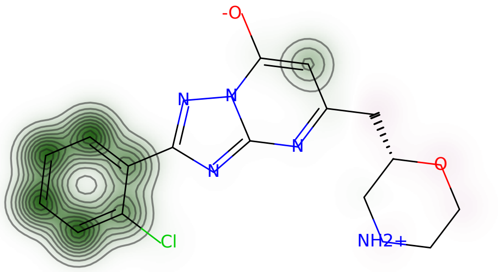
error = 1.91

ClogP = 1.07

$\mathrm{logP}_{\mathrm{pred}}$ = 2.97±0.82

overlap (ground truth) = 0.36

overlap (FPA) = 0.77



4

error = 2.68

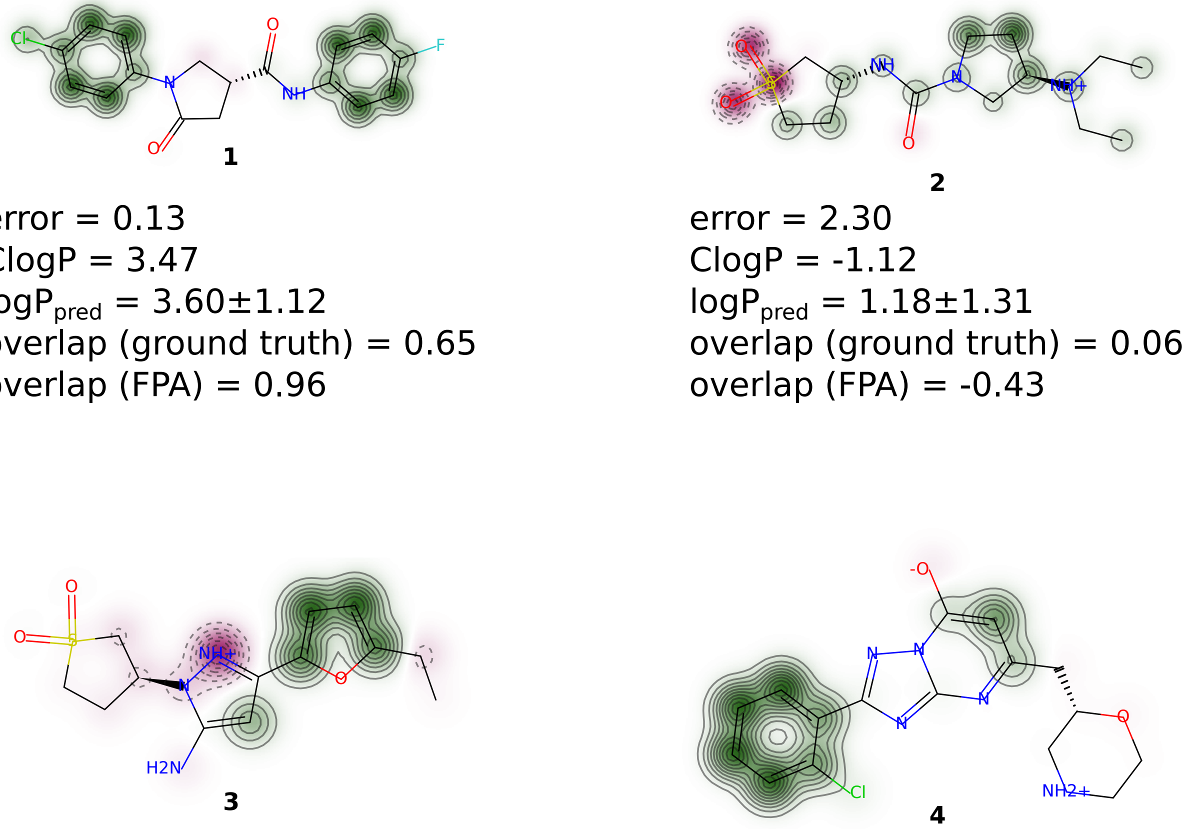
ClogP = 0.02

$\mathrm{logP}_{\mathrm{pred}}$ = 2.70±0.97

overlap (ground truth) = 0.31

overlap (FPA) = 0.79

Figure 6: ML heatmaps for molecules **1-4** for a training set size of 100 molecules.



1

error = 0.13
ClogP = 3.47
$\log P_{\text{pred}}$ = 3.60±1.12
overlap (ground truth) = 0.65
overlap (FPA) = 0.96

2

error = 2.30
ClogP = -1.12
$\log P_{\text{pred}}$ = 1.18±1.31
overlap (ground truth) = 0.06
overlap (FPA) = -0.43

3

error = 0.58
ClogP = 1.07
$\log P_{\text{pred}}$ = 1.65±1.01
overlap (ground truth) = 0.53
overlap (FPA) = 0.70

4

error = 3.37
ClogP = 0.02
$\log P_{\text{pred}}$ = 3.40±1.10
overlap (ground truth) = 0.35
overlap (FPA) = 0.75

Figure 7: ML heatmaps for molecules **1-4** for a training set size of 1000 molecules.

factor could be that the ground truth attribution must be (somehow) adapted to better reflect the molecular representation used in these graph-based models.

3.5 Results for a graph convolutional neural network

The analysis above highlights some of the challenges when using fingerprint methods for featurization of molecules in ML; specifically we saw bit collisions to be problematic for the ML model. Graph convolutional networks (GCNs) represent a means of letting the ML model learn a feature vector as an alternative to fingerprint methods. The input to the GCN (from which the feature vector can be learned) is a graph consisting of a set of node-features representing each atoms, as well as a set of edges (which atoms are bound) represented by an atom connectivity (AC) matrix with 1 on off-diagonal elements between bound atoms and 0 otherwise. The edges, which represent bonds, may also be featurized but is not considered here. The learning is typically done in an end-to-end fashion, where a neural network (NN) that ends in a prediction (here the Crippen logP) follows the GCN resulting in a graph convolutional neural network (GCNN) taking a graph as input and returning a prediction. We employ a simple GCNN using the GraphConv layer[28] as implemented in PyTorch Geometric[29, 30] and use atomic feature vectors following Dablander [31]. Details of the GCNN can be found in section S2.

Since the input to GCNN consists of a number of atomic feature vectors as well as how the atoms are connected, there is a more obvious link to getting atomic attributions compared to the fragment-based fingerprint used above. Here we use the “remove-an-atom” method used by Matveieva et al. [32]. An atom is “removed” by deleting the row with atomic features corresponding to that atom as well as all connections (edges) to that atom.

The change from ECFP4 fingerprints to learned feature vectors based on input graphs means that the error of the model decreases significantly. The 150k training set is divided into 145k for training and 5k used as validation set for early stopping. The same 5k test set is used and a RMSE of 0.16 is achieved (for RF/ECFP4 the error was 0.75).

The increased performance observed w.r.t. predictive power of the GCNN compared to the ECFP4/RF

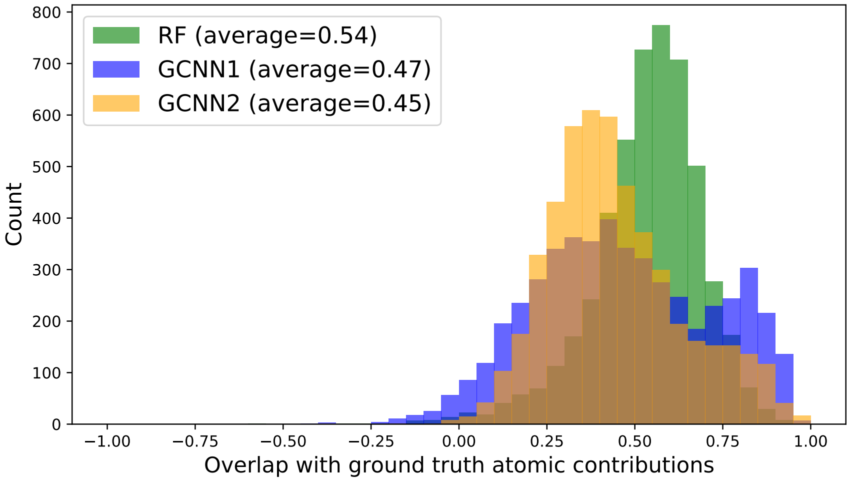


Figure 8: Distributions of overlap of the atomic attribution vectors from ML models with the ground truth atomic attributions.

model is not reflected in the agreement of the ML-generated atomic attribution heatmaps with the ground truth. The average overlap with the ground truth is 0.47 for the GCNN model (down from 0.54 for the

ECFP4/RF model). Interestingly, the distribution of overlaps is very different for the GCNN generated overlaps (labelled GCNN1 in Figure 8) compared to that of the ECFP4/RF generated overlaps (RF in Figure 8). While the ECFP4/RF overlaps show a single peak in the distribution centered roughly around the average overlap, we observe a double-peaked distribution for the GCNN overlaps with the ground truth vectors (labelled GCNN1). Thus, even though the GCNN overlaps produce a lot more atomic attribution vectors in bad agreement with the ground truth, it also produces many more atomic attribution vectors that are in close-to-perfect agreement (overlap ≈ 1) with the ground truth. The average overlap with the FPA vectors is 0.63, which is similar to the null-model where all atoms contribute the same. As the FPA vectors are designed specifically to reflect the ECFP4 fingerprint we also should not expect the GCNN generated contributions to reflect these. The average overlap with the ECFP4/RF generated atomic contributions is 0.53. While the heatmaps generated for the GCNN model are generally not closer to the ground truth than the ECFP4/RF atomic attribution heatmaps above, they might help assessing the quality of the ECFP4/RF heatmaps in the absence of the ground truth atomic attributions. We find, that the Pearson's linear correlation coefficient between the overlap of RF+ground truth and RF+GCNN is 0.56 (Figure S8a). Thus, a high overlap between atomic attributions generated by the two ML models indicates a higher probability of the ECFP4/RF attributions being in agreement with the ground truth. The correlation between GCNN+ground truth overlaps and RF+GCNN overlaps is less pronounced with a Pearson's r of 0.30 (Figure S8b).

Heatmaps of the atomic attributions from the GCNN model for the four molecules analyzed above are shown in Figure 9 (labelled GCNN1). Contrary to the heatmaps in Figure 3 for the 150k RF/ECFP4 model, the worst overlap with the Crippen atomic contributions is observed for molecule **1** (overlap = 0.20). We observe problematic behavior around the amide bond (all negative contribution on oxygen and C+N are positive) as well as on the halogens, which are predicted to make a negative contribution to logP, while the opposite is true for the ground truth contributions (Figure 3b). For molecules **2**, **3** and **4** the GCNN generated heatmaps have high overlaps (>0.75) with the ground truth.

When analyzing atomic heatmaps generated by an XAI method, we can ascribe lack of agreement with the ground truth to being caused by either the model being unknowledgeable about the atomic contributions or the XAI method being incapable of extracting those atomic contributions. For example we saw that the XAI methods used on fragment-based fingerprints above would not actually extract atomic contributions since the XAI method "removes" more than just the atom. Similarly, we can use the Crippen logP benchmark with ground truth heatmaps to understand what causes the disagreement with the GCNN generated atomic heatmaps. Qualitative assessment of the atomic GCNN heatmaps shows the "halogen" problem observed for molecule **1** to be a common problem and choose to investigate this phenomenon more thoroughly.

Figure 10 shows a heatmap of the atomic contribution for the Crippen logP model of an example molecule (**5**) where the three largest atomic contributions are from the chlorine atom and the two methyl carbon atoms. The GCNN prediction has a low error of 0.05 but the logP contribution from the chlorine atom as well as from the methyl carbon atoms are all predicted to be negative. Does that mean that the GCNN model actually assigns a negative contribution to these atoms or is it an artefact of the XAI method? If we, instead of masking the Cl atom with the XAI method, remove the Cl atom by changing it to a hydrogen atom the change in the logP predicted by the GCNN model (-0.68) is almost identical to the actual change in Crippen's logP (-0.65, Figure 11, upper arrow). This suggests that the GCNN model *has* learned about the positive contribution of a chlorine atom *and* gets its magnitude right. So why is the Cl atom given a negative contribution when we mask it with the XAI method where the logP predicted by the GCNN model increases 0.53 (Figure 11, bottom arrow). Since hydrogen atoms are not included explicitly in the graph-input, the only difference between the input where we change Cl to H and where we remove Cl with the XAI method is in the description of the carbon atom bound to Cl. Part of the atomic feature vector we use is a one hot encoding of the number of heavy-atom neighbors as well as the number of bound hydrogen atoms. The big difference in predicted logP then stems from a change from a one hot encoding of three heavy atoms to two heavy atoms and a change from zero to one bound hydrogen atoms. Thus, after masking the Cl atom with the XAI method, there still is some implicit information about it through the atomic feature vector of the bound carbon atom.

To avoid this we tried training a GCNN model without such implicit information about each atom's neighbor-

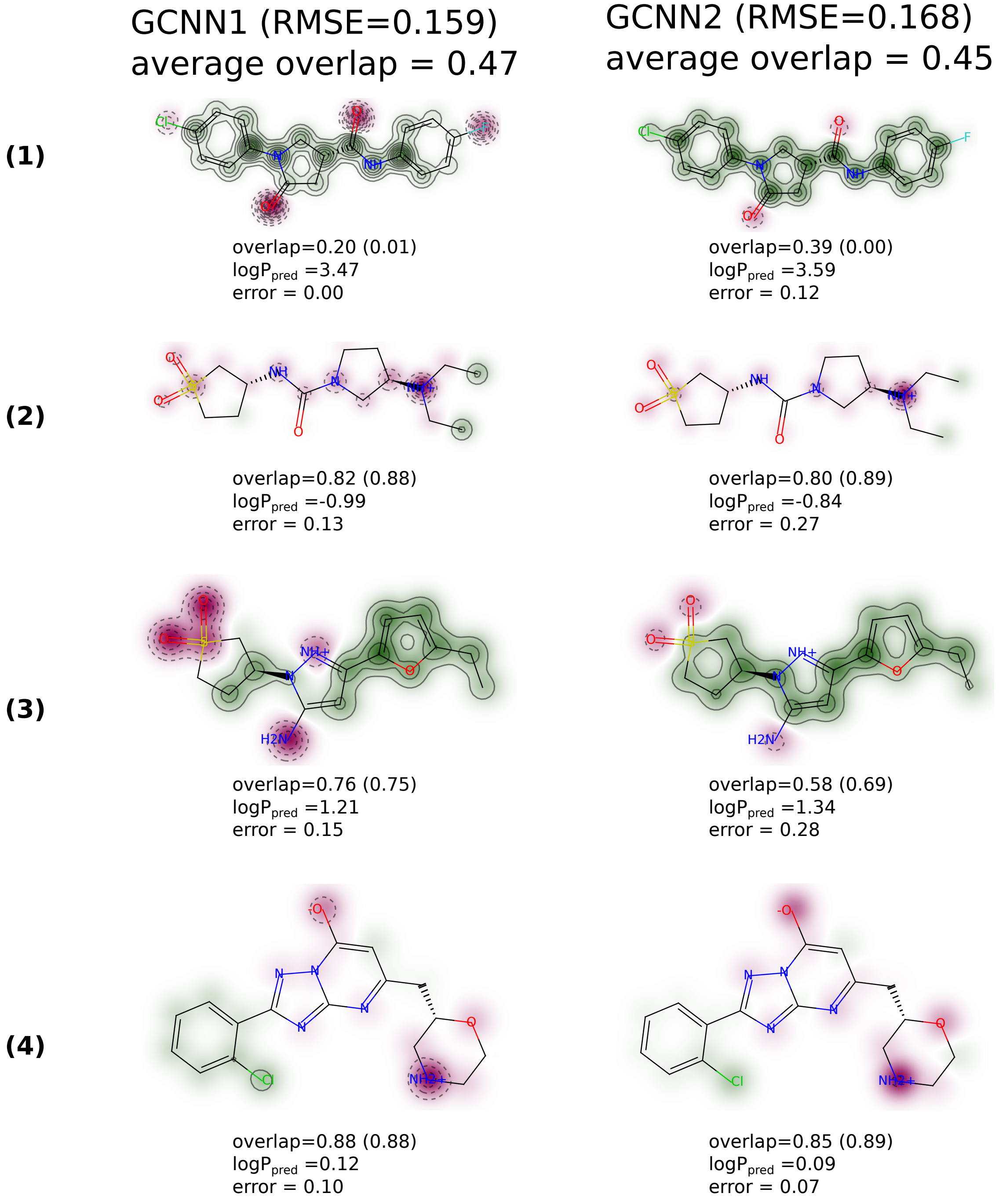


Figure 9: ML generated heatmaps of the four example molecules for the two GCNN models. Numbers in the parenthesis after overlap state the Pearson's r value.

hood, meaning that the one-hot encoding of number of heavy atom neighbors as well as number of hydrogen atoms bound are not included in the atomic feature vector (length of atomic feature vector decreases from 79 to 67). We use the same dataset as before and find the RMSE for the test set increases slightly to 0.168. Results from this model are labelled GCNN2. With this reduced set of atomic features, masking/removing

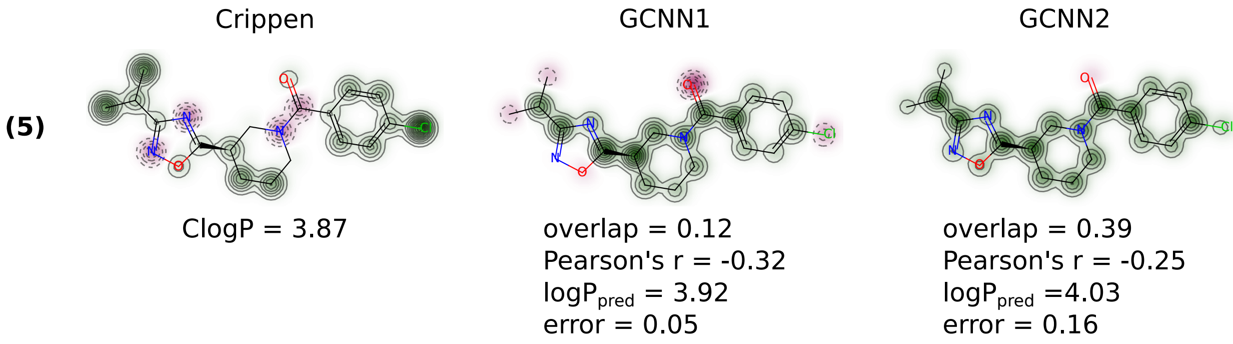


Figure 10: Heatmaps of atomic contributions for molecule **5**.

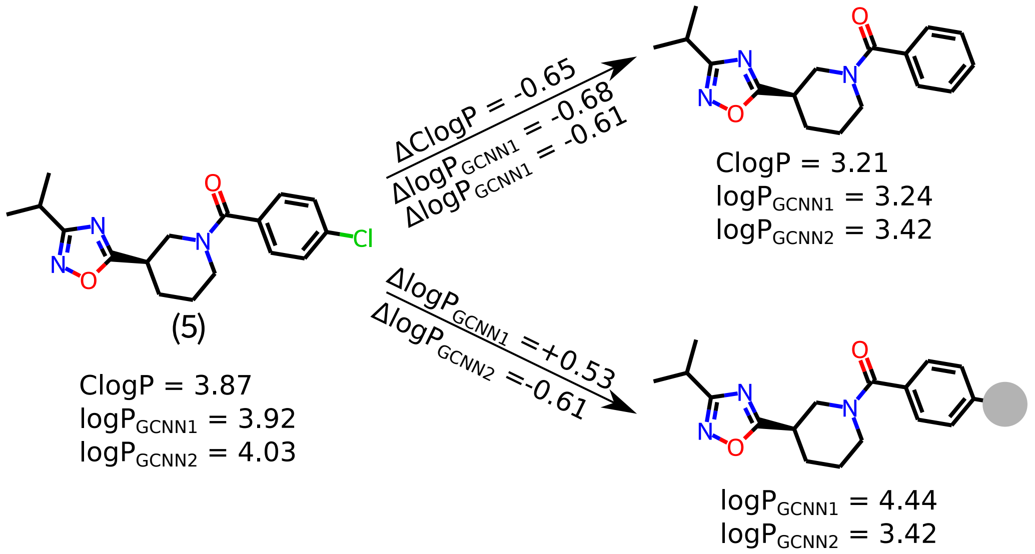


Figure 11: Effect of "removing Cl" by either changing it to a hydrogen atom (upper arrow) or masking it through the XAI procedure used (bottom arrow) as perceived by the two GCNN models.

the Cl atom with the XAI method corresponds exactly to removing it by changing it to a hydrogen atom. The change in predicted logP from the GCNN2 model is -0.61 and the Cl atom (as well as the methyl groups) are not correctly assigned a positive logP contribution (Figures 11 and 10). Note however that this does not result in a general improvement of the agreement between the ML generated atomic contributions and the ground truth, for which the average overlap decreases from 0.47 to 0.45 (Figure 8). For the four example molecules we generally observe heatmaps that are similar to those generated by the GCNN1 model 9).

This section again illustrates how the quantitative benchmark for regression in the form of atomic contributions from the Crippen logP model can be used for probing specific behavior of an XAI method + ML model combination. In particular, we saw how the benchmark set was used to identify problematic behavior w.r.t. how the XAI method evaluated some of the atomic contributions. This type of error probing is crucial in order to improve XAI methods applied to chemistry. Furthermore, we saw how one can not assume better XAI performance simply from working with a more accurate model; explainability and model performance is not necessarily correlated.

4 Conclusions and outlook

The “atom attribution from finger prints”-method developed by Riniker and Landrum, when used with a RF model fitted to Crippen logP values, gives atomic attribution heatmaps that are in reasonable agreement with the atomic contribution heatmaps of the Crippen logP model for most molecules, with average heatmap overlaps of up to 0.54. The agreement is increased significantly (to 0.75) when the atomic contributions are adjusted to match the fact that the molecular representation (FPs) is fragment-based rather than atom-based (the FPA ground truth vector). Molecules where the atomic attribution differs significantly from the ground truth tend to have slightly larger errors on average, but the correlation factor is near zero when considering individual molecules.

Most heatmaps and the corresponding FPA overlaps are relatively insensitive to the training set size and the results are close to converged for a training set size of 1000 molecules, although for molecules with low overlap some heatmaps change significantly. The difference in average error for molecules with high and low overlap is more pronounced for smaller training sets (Figure S7), but this could just reflect the larger spread in errors for models trained on smaller training sets.

Using a GCNN with the XAI method suggested by Maveieva et al.[32] we generally found a worse agreement with the ground truth heatmaps reflected in an average heatmap overlap of 0.47.

The Crippen logP model was used to reveal problematic behaviur of two XAI methods suggested for molecules; the dummy atom approach for fingerprints [4] and the “remove an atom” approach for GCNNs[32]. At first sight both methods seem to be reasonable ways of “removing an atom”, demonstrating the importance of having quantitative regression benchmarks for XAI revealing unforeseen problems.

Our main conclusion is that the Crippen logP model provides an excellent benchmark for heatmap approaches, especially if the ground truth heatmaps can be adjusted to reflect the molecular representation. While this is straightforward for a FP representation, it is not immediately clear how to do this for graph-based representations. In any case, we have shown that a combination of a relatively simple and widely used ML model and attribution method can provide heatmaps that are in good agreement with the ground truth and give a great deal of insight into how the model has learned. Like the simpler attribution benchmarks for classification tasks that have come before it, this work sets the bar for regression tasks.

References

[1] Lior Hirschfeld, Kyle Swanson, Kevin Yang, Regina Barzilay, and Connor W Coley. "Uncertainty Quantification Using Neural Networks for Molecular Property Prediction". In: *J. Chem. Inf. Model.* 60.8 (Aug. 2020), pp. 3770–3780. DOI: 10.1021/acs.jcim.0c00502.

[2] Pavel Polishchuk. "Interpretation of Quantitative Structure-Activity Relationship Models: Past, Present, and Future". In: *J. Chem. Inf. Model.* 57.11 (Nov. 2017), pp. 2618–2639. DOI: 10.1021/acs.jcim.7b00274.

[3] Sereina Riniker and Gregory A Landrum. "Similarity maps - a visualization strategy for molecular fingerprints and machine-learning methods". In: *J. Cheminform.* 5.1 (Sept. 2013), p. 43. DOI: 10.1186/1758-2946-5-43.

[4] Robert P Sheridan. "Interpretation of QSAR Models by Coloring Atoms According to Changes in Predicted Activity: How Robust Is It?" In: *J. Chem. Inf. Model.* 59.4 (Apr. 2019), pp. 1324–1337. DOI: 10.1021/acs.jcim.8b00825.

[5] Benjamin Sanchez-Lengeling, Jennifer Wei, Brian Lee, Emily Reif, Peter Wang, Wesley Qian, Kevin McCloskey, Lucy Colwell, and Alexander Wiltschko. "Evaluating Attribution for Graph Neural Networks". In: *Advances in Neural Information Processing Systems.* Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 5898–5910. URL: https://proceedings.neurips.cc/paper/2020/file/417fbbf2e9d5a28a855a11894b2e795a-Paper.pdf.

[6] Kevin McCloskey, Ankur Taly, Federico Monti, Michael P Brenner, and Lucy J Colwell. "Using attribution to decode binding mechanism in neural network models for chemistry". In: *Proc. Natl. Acad. Sci. U. S. A.* 116.24 (June 2019), pp. 11624–11629. DOI: 10.1073/pnas.1820657116.

[7] Henry Heberle, Linlin Zhao, Sebastian Schmidt, Thomas Wolf, and Julian Heinrich. "XSMILES: interactive visualization for molecules, SMILES and XAI attribution scores". Oct. 2022. DOI: 10.21203/rs.3.rs-2121152/v1.

[8] Pavel G Polishchuk, Victor E Kuz'min, Anatoly G Artemenko, and Eugene N Muratov. "Universal Approach for Structural Interpretation of QSAR/QSPR Models". In: *Mol. Inform.* 32.9-10 (Oct. 2013), pp. 843–853. DOI: 10.1002/minf.201300029.

[9] Geemi P Wellawatte, Aditi Seshadri, and Andrew D White. "Model agnostic generation of counterfactual explanations for molecules". In: *ChemRxiv* (Sept. 2021). DOI: 10.26434/chemrxiv-2021-4qkg8-v2.

[10] Ana Lucic, Maartje ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. "CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks". In: *arXiv* (Feb. 2021). DOI: 10.48550/arXiv.2102.03322.

[11] Guang-He Lee, Wengong Jin, David Alvarez-Melis, and Tommi S Jaakkola. "Functional Transparency for Structured Data: a Game-Theoretic Approach". In: *arXiv* (Feb. 2019). DOI: 10.48550/arXiv.1902.09737.

[12] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. "GNNExplainer: Generating Explanations for Graph Neural Networks". In: *arXiv* (Mar. 2019). DOI: 10.48550/arXiv.1903.03894.

[13] Tobias Harren, Hans Matter, Gerhard Hessler, Matthias Rarey, and Christoph Grebner. "Interpretation of Structure-Activity Relationships in Real-World Drug Design Data Sets Using Explainable Artificial Intelligence". In: *J. Chem. Inf. Model.* 62.3 (Feb. 2022), pp. 447–462. DOI: 10.1021/acs.jcim.1c01263.

[14] Ryan Henderson, Djork-Arné Clevert, and Floriane Montanari. "Improving Molecular Graph Neural Network Explainability with Orthonormalization and Induced Sparsity". In: *arXiv* (May 2021). DOI: 10.48550/arXiv.2105.04854.

[15] José Jiménez-Luna, Miha Skalic, Nils Weskamp, and Gisbert Schneider. "Coloring Molecules with Explainable Artificial Intelligence for Preclinical Relevance Assessment". In: *J. Chem. Inf. Model.* 61.3 (Mar. 2021), pp. 1083–1094. DOI: 10.1021/acs.jcim.0c01344.

[16] José Jiménez-Luna, Francesca Grisoni, and Gisbert Schneider. "Drug discovery with explainable artificial intelligence". In: *Nature Machine Intelligence* 2.10 (Oct. 2020), pp. 573–584. DOI: 10.1038/s42256-020-00236-4.

[17] José Jiménez-Luna, Miha Skalic, and Nils Weskamp. "Benchmarking Molecular Feature Attribution Methods with Activity Cliffs". In: *J. Chem. Inf. Model.* 62.2 (Jan. 2022), pp. 274–283. DOI: 10.1021/acs.jcim.1c01163.

[18] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70.* ICML'17. Aug. 2017, pp. 3319–3328. URL: https://proceedings.mlr.press/v70/sundararajan17a.html.

[19] John S Delaney. "ESOL: estimating aqueous solubility directly from molecular structure". In: *J. Chem. Inf. Comput. Sci.* 44.3 (May 2004), pp. 1000–1005. DOI: 10.1021/ci034243x.

[20] Scott A Wildman and Gordon M Crippen. "Prediction of Physicochemical Parameters by Atomic Contributions". In: *J. Chem. Inf. Comput. Sci.* 39.5 (Sept. 1999), pp. 868–873. DOI: 10.1021/ci990307l.

[21] Greg Landrum. *RDKit: Open-source cheminformatics.* 2020. URL: http://www.rdkit.org.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[23] Xiufeng Yang, Jinzhe Zhang, Kazuki Yoshizoe, Kei Terayama, and Koji Tsuda. "ChemTS: an efficient python library for de novo molecular generation". In: *Sci. Technol. Adv. Mater.* 18.1 (Nov. 2017), pp. 972–976. DOI: 10.1080/14686996.2017.1401424.

[24] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules". In: *ACS Cent Sci* 4.2 (Feb. 2018), pp. 268–276. DOI: 10.1021/acscentsci.7b00572.

[25] Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. "Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation". In: *arXiv* (June 2018). DOI: 10.48550/arXiv.1806.02473.

[26] Jan H Jensen. "A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space". en. In: *Chem. Sci.* 10.12 (2019), pp. 3567–3572.

[27] David Rogers and Mathew Hahn. "Extended-connectivity fingerprints". In: *J. Chem. Inf. Model.* 50.5 (May 2010), pp. 742–754.

[28] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. "Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks". In: *AAAI* 33.01 (July 2019), pp. 4602–4609. DOI: 10.1609/aaai.v33i01.33014602.

[29] A. Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32.* Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035. DOI: 10.5555/3454287.3455008.

[30] Matthias Fey and Jan E. Lenssen. "Fast Graph Representation Learning with PyTorch Geometric". In: *ICLR Workshop on Representation Learning on Graphs and Manifolds.* 2019.

[31] M. Dablander. *How to turn a SMILES string into a molecular graph for Pytorch Geometric*. Blog post. Feb. 2022. URL: `https://www.blopig.com/blog/2022/02/how-to-turn-a-smiles-string-into-a-molecular-graph-for-pytorch-geometric/#more-7803`.

[32] Mariia Matveieva and Pavel Polishchuk. “Benchmarks for interpretation of QSAR models”. In: *Journal of cheminformatics* 13.1 (May 2021), p. 41. DOI: `10.1186/s13321-021-00519-x`.

Supporting Information

The code and data resulting from this study can be found here https://github.com/jensengroup/FP_RF_XAI and https://sid.erda.dk/sharelink/eUVFpTDU62, respectively.

Table S1: Table of training and test MAE values

			Training set size								
			100	500	1000	5000	10.000	20.000	50.000	100.000	150.000
N_{bits}	2048	train	0.59	0.46	0.44	0.39	0.37	0.36	0.34	0.32	0.31
		test	1.01	0.86	0.79	0.73	0.71	0.67	0.63	0.60	0.58
	1024	train	0.57	0.45	0.44	0.39	0.38	0.37	0.35	0.33	0.32
		test	1.03	0.88	0.82	0.76	0.74	0.71	0.66	0.63	0.62

Table S2: Mean of Pearson's r for the test set

	Training set size			
	100	500	1.000	150.000
FPA	0.63	0.66	0.69	0.74
“ground truth”	0.34	0.39	0.43	0.46

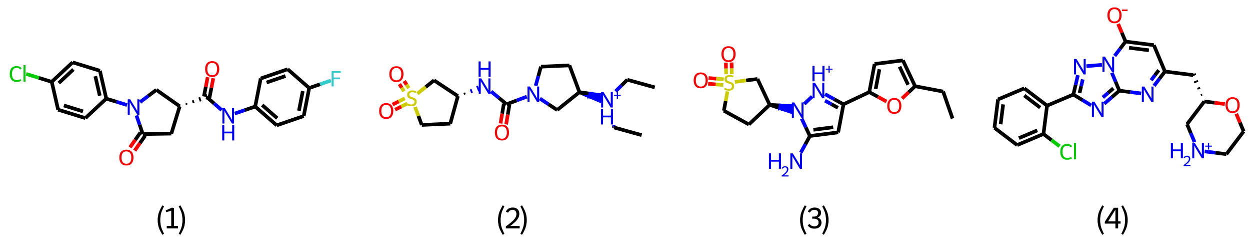


Figure S1: The four molecules investigated in detail in this work.

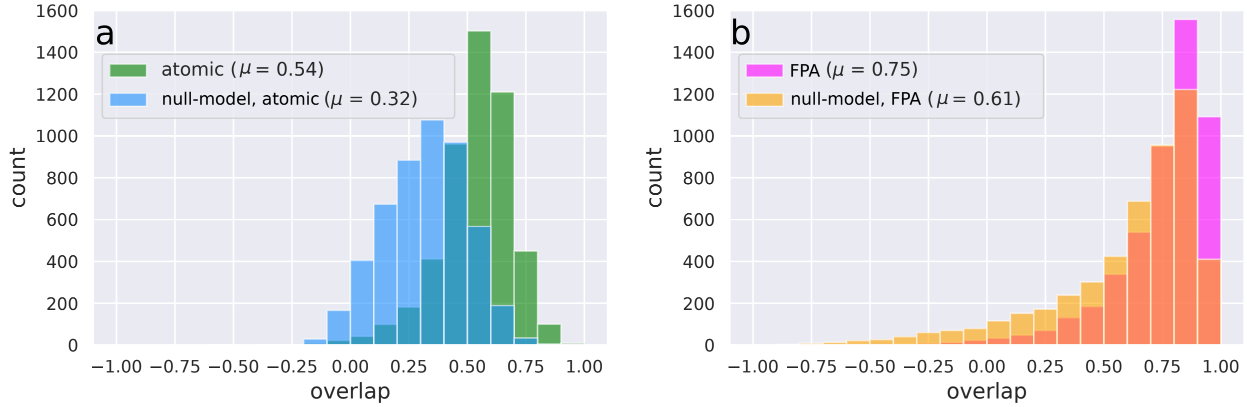


Figure S2: overlap distributions for a) the atomic-attribution vector and the ML vector (green) compared to the overlap between the atomic attribution vector and the null-model vector (blue) and b) the FP-attribution vector and the ML vector (magenta) compared to the overlap between the FP-attribution vector and the zero-model vector (orange). The null-model vector is created by simply assigning the mean atom contribution to each atom based on the predicted logP; for positive logP predictions all atoms will have the same positive contribution and for negative logP predictions all atom will have the same negative contribution.

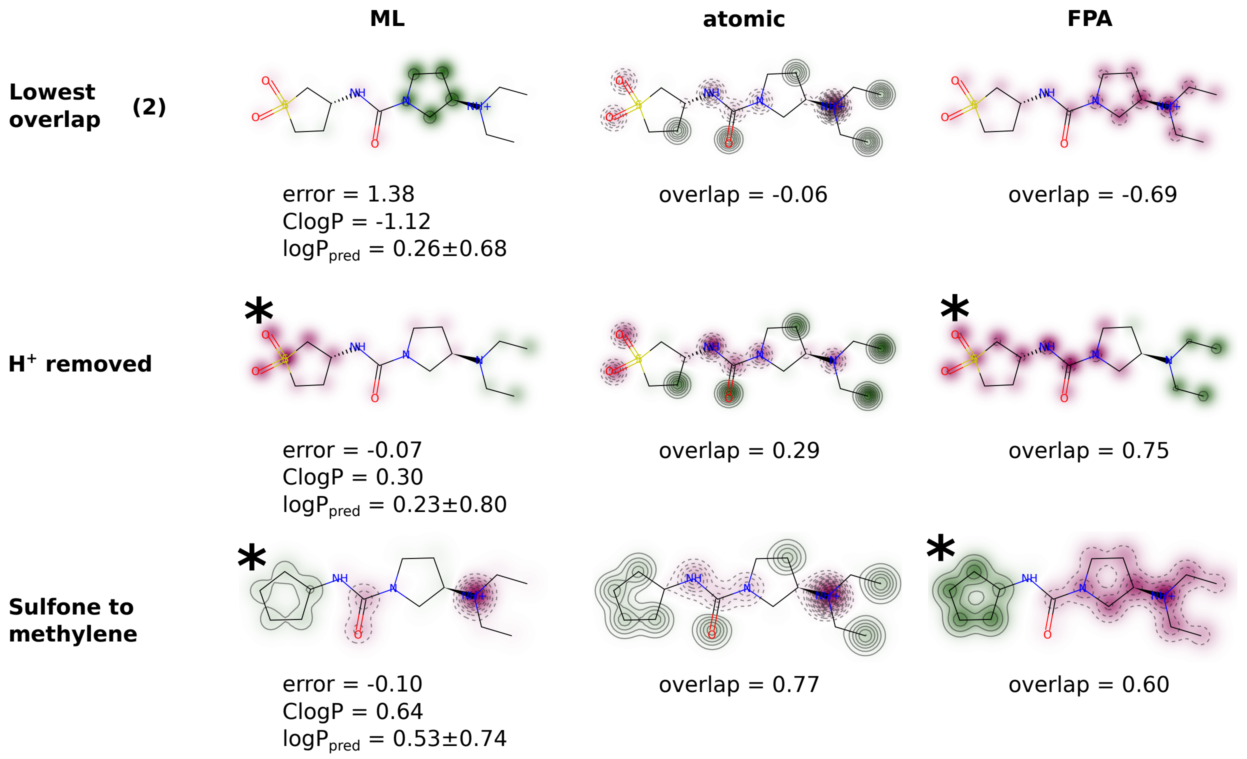


Figure S3: Effect of removing proton as well as changing a sulfone group to a methyl group from the molecule with lowest FPA overlap. Heatmaps with sign problems are marked with *.

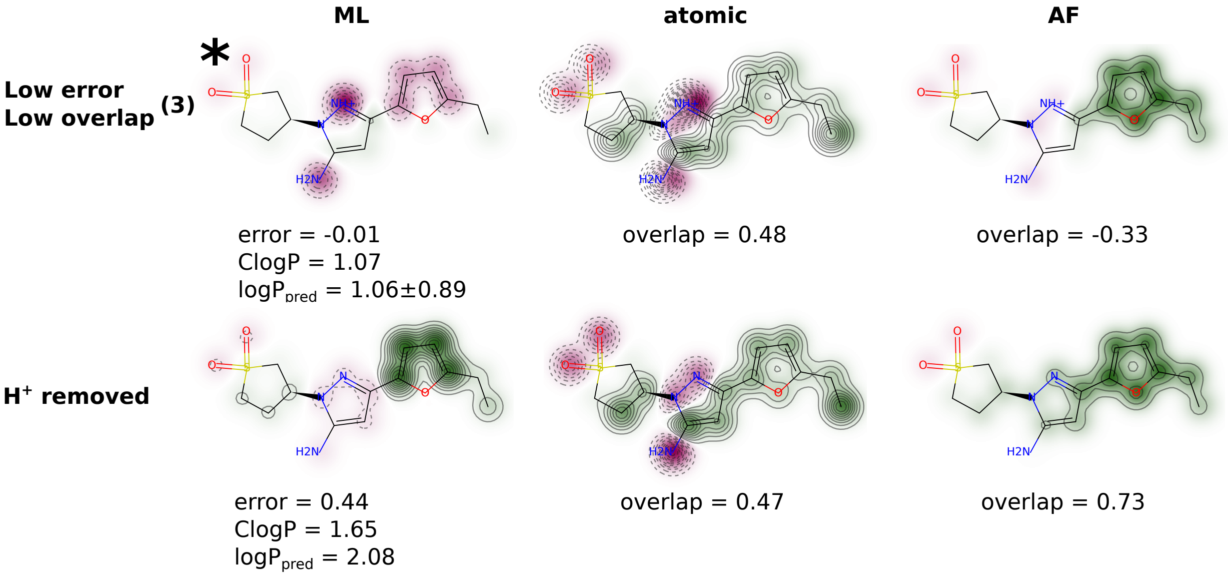


Figure S4: Effect of removing proton from a molecule with low FPA overlap and low error.

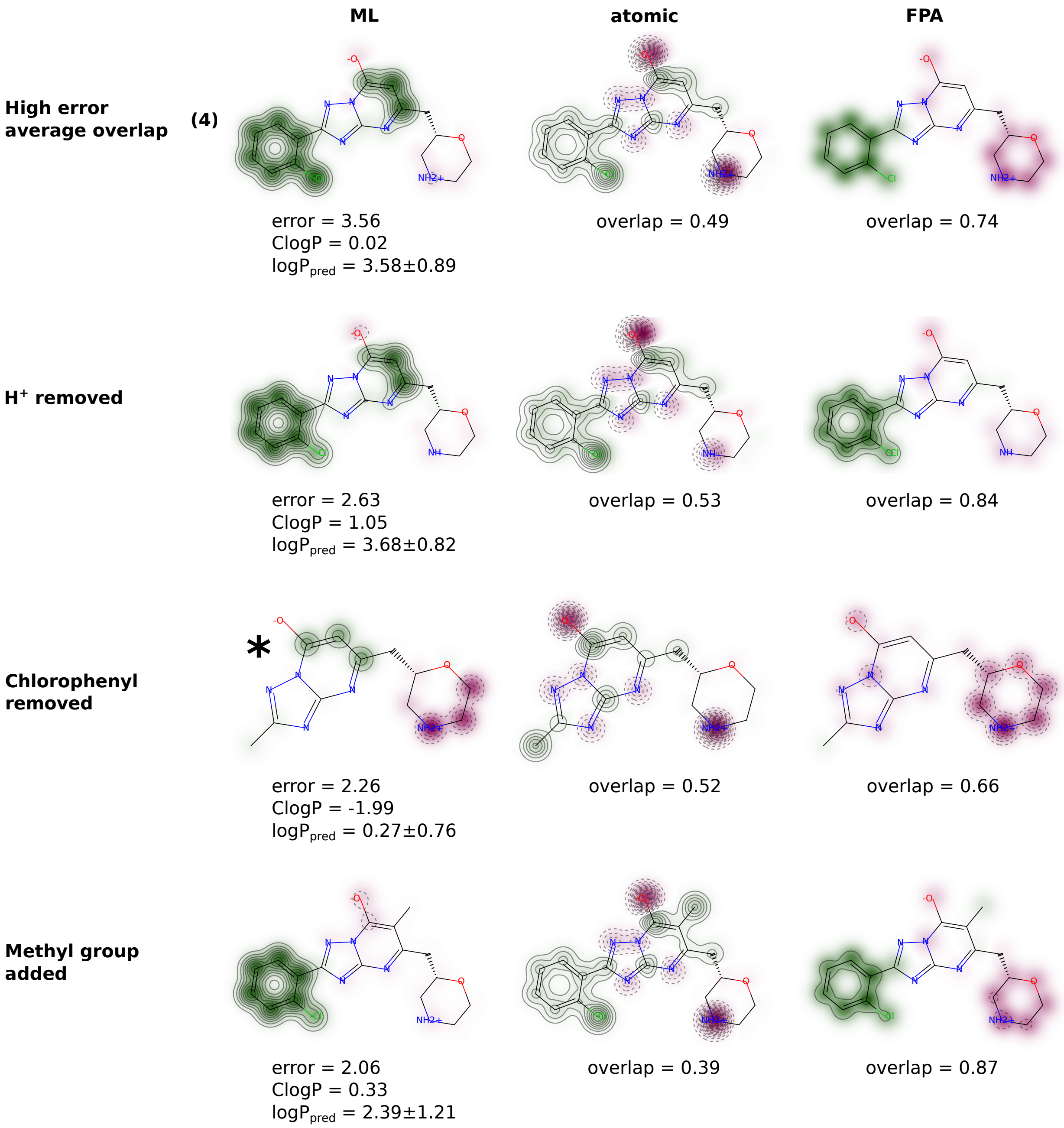


Figure S5: Effect of removing proton from a molecule with high FPA overlap and low error.

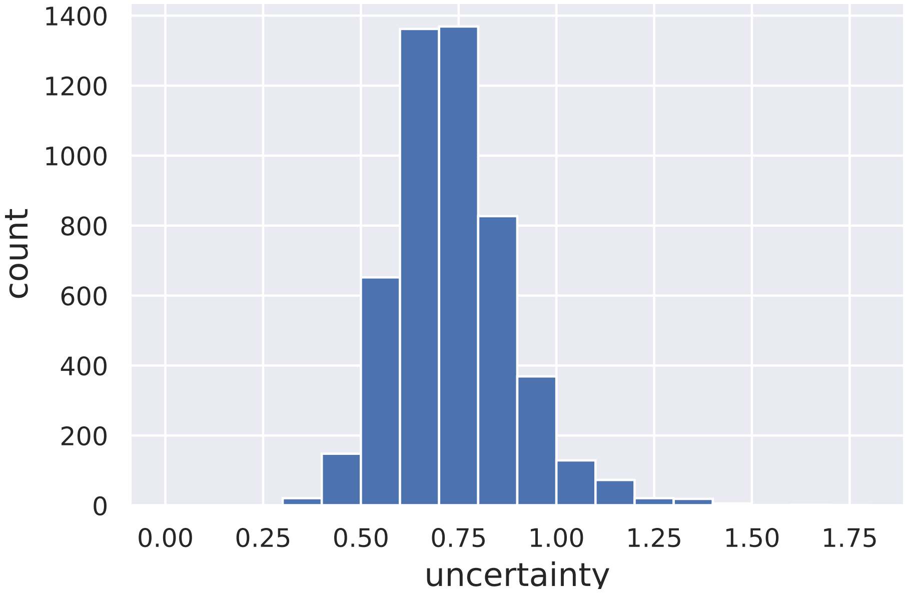


Figure S6: Distribution of predicted uncertainties for the test set with the model trained on 150.000 ClogP values.

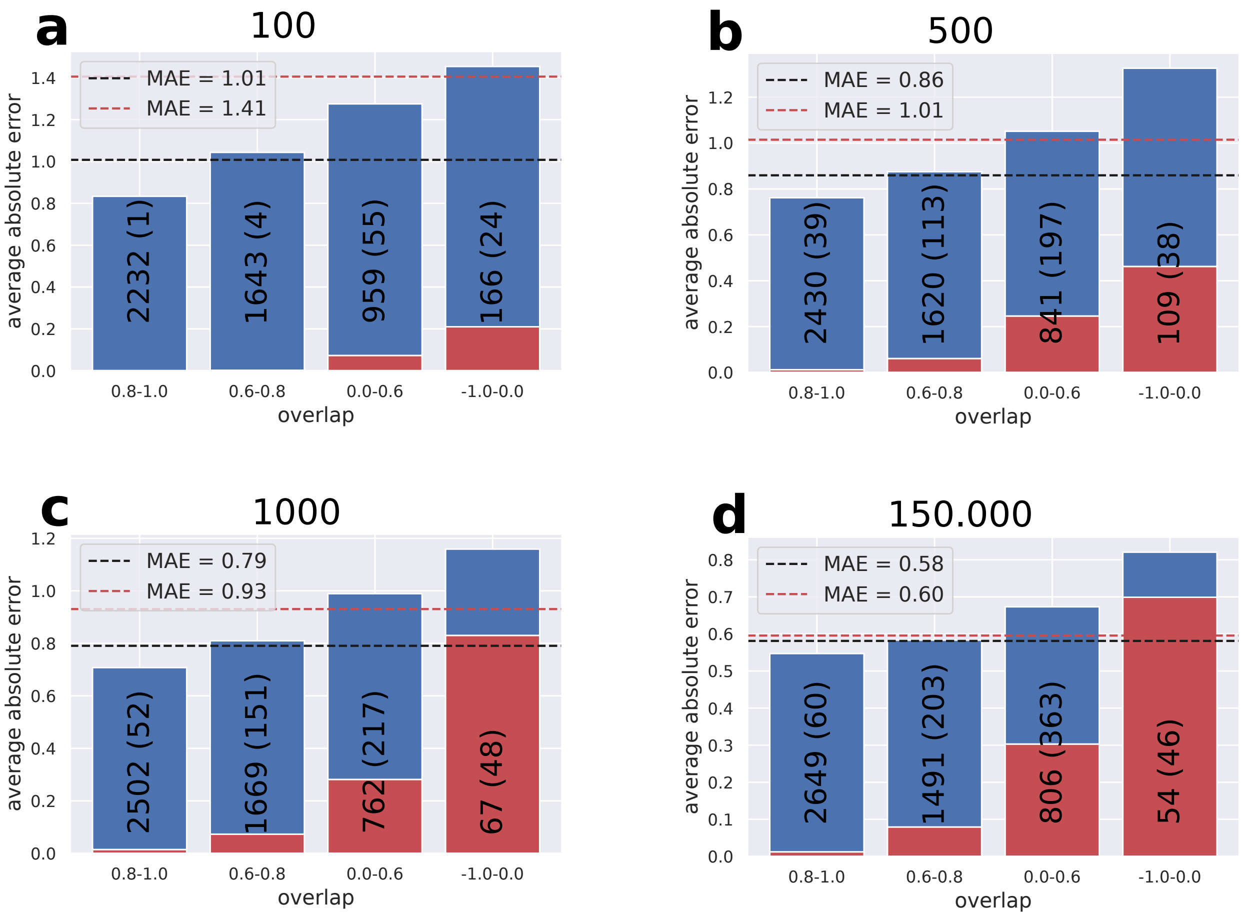


Figure S7: Overlap, error and sign problem analysis on test set based on model trained on 100, 500 and 150.000 entries with bit vector of length 2048. Numbers on the bar state the total number of entries and the number in parenthesis how many of these entries that have the “sign problem”. Doing the same analysis on the first 5000 entries of the training set for the 150.000 model results in an increased number of molecules with sign problem (845 vs 672 for the test set). Doing the same analysis on the test set but with a model trained on 150.000 and with a bit vector length of 1024 results in a decrease in the total number of molecules with the sign problem (558 vs. 672).

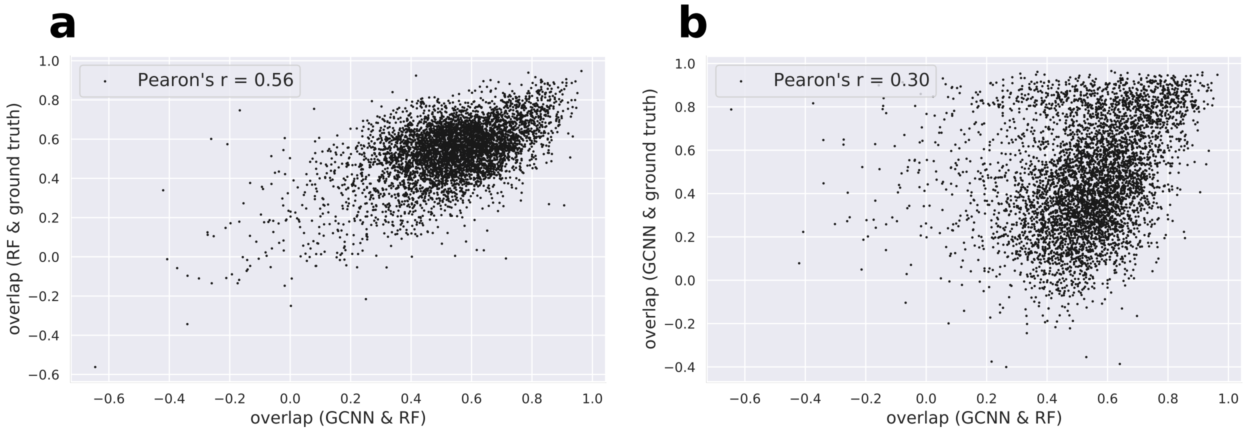
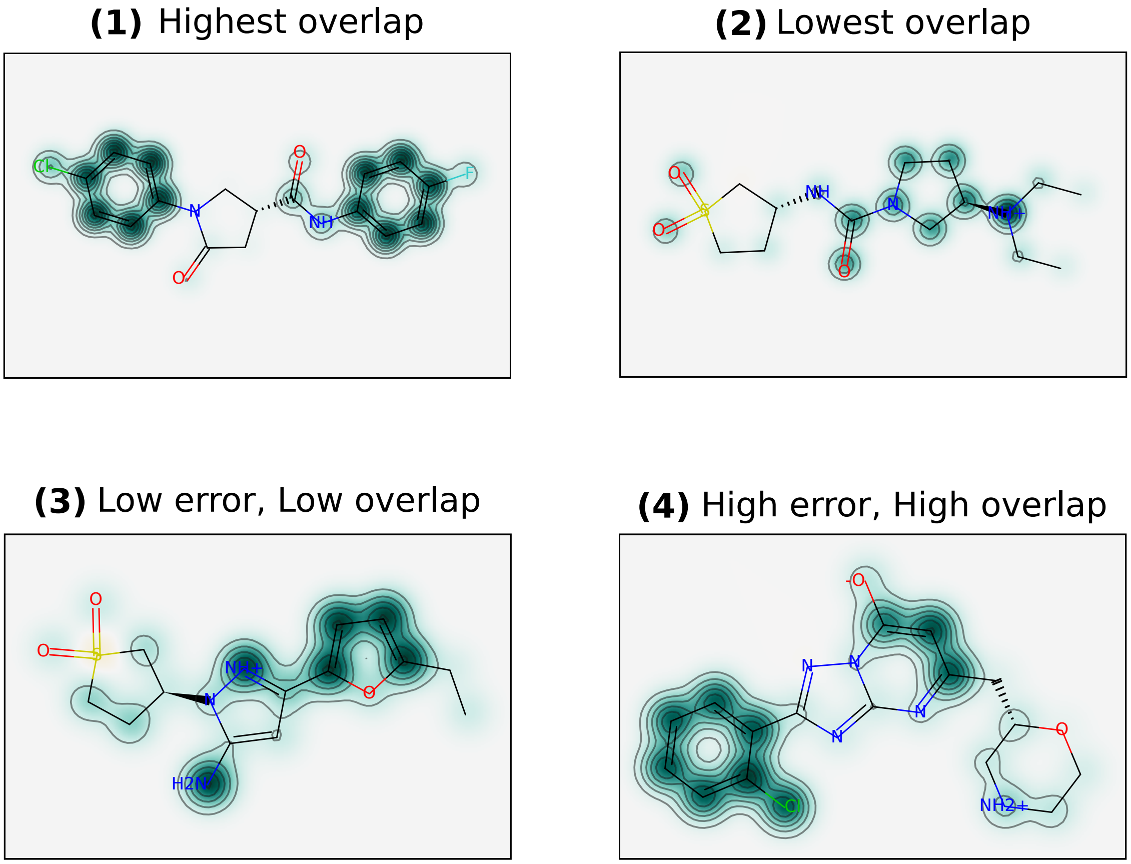


Figure S8: a) Overlap between the atomic attributions from the 150k ECFP4/RF model and the ground truth atomic contributions (Crippen) vs. overlap between the atomic attributions from the 150k ECFP4/RF model and atomic attributions from the 145k+5k GCNN model. b) Overlap between the atomic attributions from the 145k+5k GCNN model and the ground truth atomic contributions (Crippen) vs. overlap between the atomic attributions from the 150k ECFP4/RF model and atomic attributions from the 145k+5k GCNN model. Evaluated on the 5k test set.

S1 Uncertainty heatmaps

The two different approaches for generating uncertainty-related heatmaps, UAA and AAU, target different kinds of atomic uncertainties. The first, uncertainty in atomic attributions (UAA), assesses how uncertain the model is in what attribution to give to each atom. Since the RF model is an ensemble model, we can in principle obtain a logP-heatmap from each tree in the model (in this case 200). We then use the standard deviation of the 200 predicted atomic attributions as a measure of how uncertain the model is of the atomic attribution. On the other hand, the atomic attribution to the uncertainty (AAU) aims at estimating how much each atom contributes to the overall uncertainty in the predicted logP. Thus, when we ”remove an atom” we assign the difference in the uncertainty prediction to that atom as the atom’s contribution to the uncertainty.


(1) Highest overlap

(2) Lowest overlap

(3) Low error, Low overlap

(4) High error, High overlap

Figure S9: UAA uncertainty heatmaps of the four molecules depicted in Figure 3. Depicted uncertainties are scaled such that 0.3 is the zero-point (no color or contour on the map) and the color scale is with respect to the maximum atom uncertainty in the molecule. Each contour represents a 0.2 increase in uncertainty.

Figure S9 shows heatmaps of the UAA for the four molecules discussed so far, which generally emphasize the same parts of the molecules as the logP attributions, i.e. the atoms that contribute most to the logP value tend to have the highest uncertainties. However, the heatmaps for molecule **2** and **4** do indicate uncertainties for the cationic atoms that are higher than the corresponding logP contributions and similarly for the C atoms in the furan ring of **3**. The AAU plots show a reverse trend where the highest uncertainties often are in regions that make relatively small logP contributions. The plots for molecules **3** and **4** show the highest uncertainties for the cationic atoms as well as the C atoms in the furan ring for **3**. Interestingly, for molecule **2** the S atom of the sulfone group contributes significantly to the uncertainty in the logP prediction.

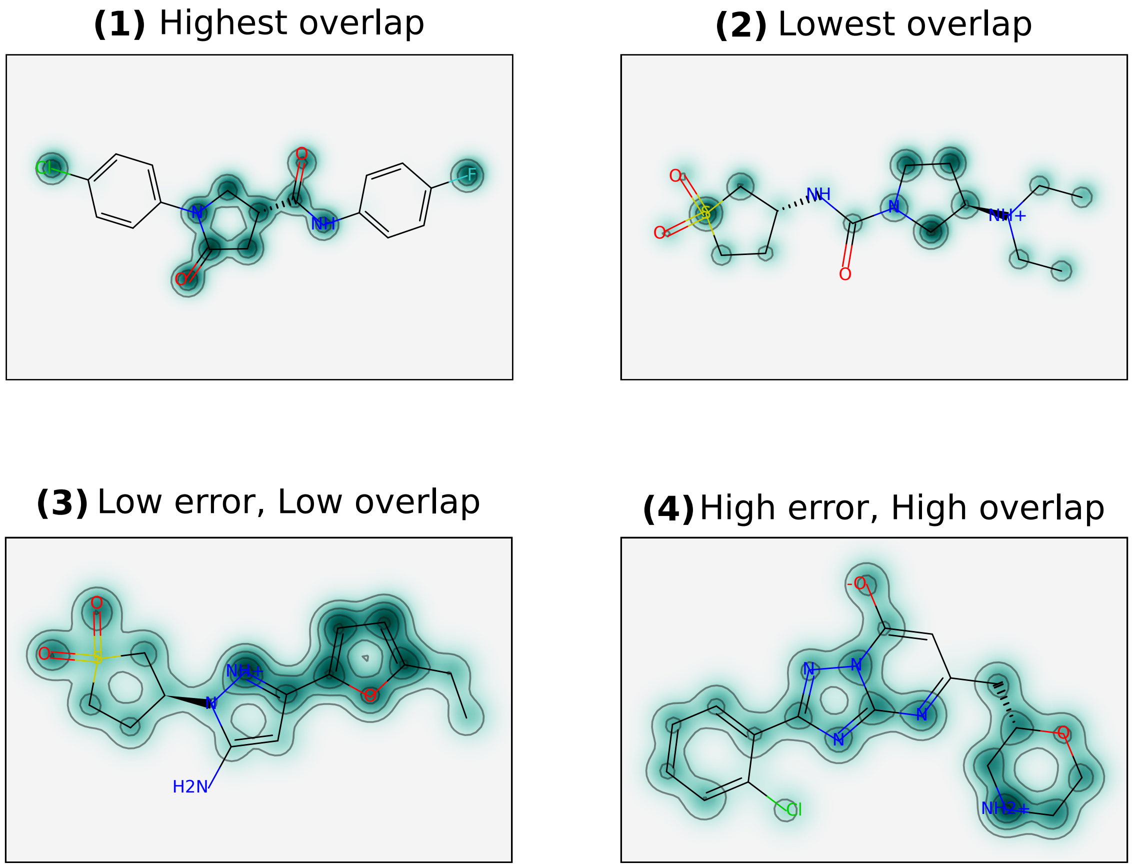


Figure S10: AAU uncertainty heatmaps of the predicted uncertainty on logP. The plots are scaled so that the lowest atom contribution set to 0 and the atom contributions sum to the total uncertainty on the predicted logP. Each contour represent a 0.02 difference.

To summarise, the plots of the UAA and AAU can help identify molecular regions that contribute significantly to errors in the logP prediction and/or attribution and can be used to guide the design of counterfactual examples to probe the ML model further (Figures S3-S5).

S2 Model details

GCNN (PyTorch Geometric)

- Input atom feature vector; 1-hot encoding of atomic number, hybridization, formal charge, chirality, number of hydrogen neighbors and number of heavy atom neighbors. Binary encoding of whether atom is in ring of not and if it aromatic or not. Scaled numeric values of atomic mass, van der Waals radius and covalent radius. Length of atomic feature vector: 79
- Three GCN layers (GraphConv) each with 300 hidden nodes and with ReLU activation functions.
- Pooling layer combining the atomic feature vectors by calculating the average producing a learned molecular feature vector of size 300.

- A feed-forward NN following the pooling layer with three hidden layers each consisting of 300 nodes in each, with ReLU activation functions and a dropout rate of 0.2.
- Learning rate = 10^{-4}
- Batch size = 128
- Loss = mean squared error (MSE)
- Early stopping with a patience of 30 epochs is used.

Feed-forward NN (PyTorch)

- Input molecular features: revised auto-correlations (RACs[**Janet2017-lg**]) in addition to the oxidation state, spin multiplicity and total charge of the ligands. RACs that are invariant over the dataset are eliminated leaving 154 input features.
- Both input features and output targts are scaled according to the training data using the StandardScaler of scikit-learn.
- Three hidden layers with 300 nodes in each, ReLU activation and dropout rate of 0.2.
- Output layer with a single node and no activation function.
- Batch size = 64
- Loss = MSE
- Learning rate = 10^{-4}
- Early stopping with a patience of 30 epochs is used.