GCI: A (G)RAPH (C)ONCEPT (I)NTERPRETATION FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Explainable AI (XAI) underwent a recent surge in research on concept extraction, focusing on extracting human-interpretable concepts from Deep Neural Networks. An important challenge facing concept extraction approaches is the difficulty of interpreting and evaluating discovered concepts, especially for complex tasks such as molecular property prediction. We address this challenge by presenting GCI: a (G)raph (C)oncept (I)nterpretation framework, used for quantitatively measuring alignment between concepts discovered from Graph Neural Networks (GNNs) and their corresponding human interpretations. GCI encodes concept interpretations as functions, which can be used to quantitatively measure the alignment between a given interpretation and concept definition. We demonstrate four applications of GCI: (i) quantitatively evaluating concept extractors, (ii) measuring alignment between concept extractors and human interpretations, (iii) measuring the completeness of interpretations with respect to an end task and (iv) a practical application of GCI to molecular property prediction, in which we demonstrate how to use chemical *functional groups* to explain GNNs trained on molecular property prediction tasks, and implement interpretations with a 0.76 AUCROC completeness score.

1 INTRODUCTION

Graph Neural Networks (GNNs) have emerged as a powerful paradigm for Deep Learning on graphs, and have been applied in a wide variety of domains including social, biological, physical, and chemical (Sanchez-Lengeling et al., 2021; Zhou et al., 2020; Wu et al., 2020; Scarselli et al., 2008). Similarly to other Deep Learning approaches, such as those based on Convolutional Neural Networks (CNNs), GNNs are black-box models whose behaviour cannot be interpreted directly.

Consequently, recent research has attempted to address this issue by applying existing Explainable AI approaches to GNNs Yuan et al. (2022); Pope et al. (2019); Ying et al. (2019). The most widely used explainability approaches for applied to GNNs are feature importance methods Ying et al. (2019). For a given data point, these methods provide scores that show the importance of each feature (i.e., subgraph, node, or edge) to the algorithm's decision. Unfortunately, these methods have been shown to be fragile to input Ghorbani et al. (2019a); Kindermans et al. (2019) or model parameter Adebayo et al. (2018); Dimanov et al. (2020) perturbations. Human experiments also demonstrate that feature importance explanations do not necessarily increase human understanding, trust, or ability to correct mistakes in a model Poursabzi-Sangdeh et al. (2018); Kim et al. (2018).

As a consequence, more recent approaches to GNN explainability have focused on *concept-based* explanations (Magister et al., 2021; 2022; Georgiev et al., 2022). Concept-based approaches aim to provide explanations of a Deep Neural Network (DNN) model in terms of human-understandable units, rather than individual features, pixels, or characters Zhou et al. (2018); Ghorbani et al. (2019b); Dimanov (2021); Zarlenga et al. (a); Kazhdan et al. (2020a). In particular, *concept extraction* approaches (a subfield of concept-based explanations) aim to extract interpretable concepts from Deep Learning models, in order to better understand which concepts a model has learned, and see if a model has picked up novel concepts that could improve existing domain knowledge Ghorbani et al. (2019b); Magister et al. (2021).

A key challenge for concept extraction approaches is that they are predominantly evaluated using qualitative methods, such as visually inspecting whether an extracted concept contains groups of superpixels that are semantically meaningful (e.g., 'these superpixels all contain a wheel, therefore the likely concept is *wheel*') (Kazhdan et al., 2020b; Ghorbani et al., 2019b; Kim et al., 2017; Zarlenga et al., a). However, this qualitative approach makes it difficult to assess which interpretations are more/less accurate. Furthermore, this makes it difficult to compare different concept extraction approaches to each other.

We address the above challenge by presenting GCI: a (G)raph (C)oncept (I)nterpretation framework[1], used for *quantitatively* verifying concepts extracted from GNNs, using provided human interpretations.

In summary, we make the following contributions:

1. We present GCI: (G)raph (C)oncept (I)nterpretation, the first framework capable of quantitatively analysing extracted concepts for GNNs on graph classification tasks;
2. Using several synthetic case-studies, we demonstrate how GCI can be used for (i) quantitatively evaluating the quality of concept extractors, (ii) measuring alignment between different concept extractors and human interpretations, (iii) measuring the completeness of interpretations with respect to an end task;
3. We demonstrate how GCI can be used for molecular property prediction tasks, by using a Blood-Brain Barrier Penetration task, and encoding *functional groups* as concepts. In particular, we show how GCI can be used to explain a GNN model trained on this molecular property prediction task, and implement interpretations achieving a 0.76 AUCROC completeness score.

2 Methodology

In this section we present our GCI approach, describing how it can be used for quantitatively analysing extracted concepts. A visual summary of our GCI approach is shown in Figure 1.

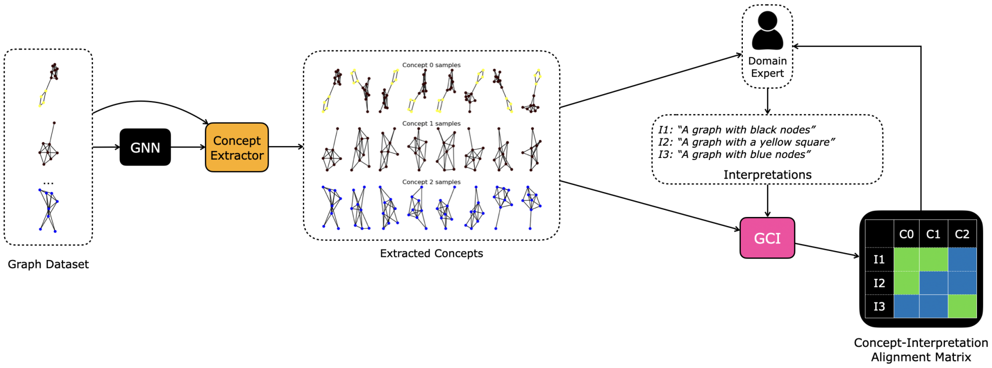


Figure 1: GCI framework overview. Left-to-right: Firstly, a Concept Extractor extracts a set of concepts from a GNN trained on a graph dataset. Next, these concepts are observed by a domain expert. The domain expert then generates a set of *interpretations* of the generated concepts. These interpretations and concepts are fed into the GCI framework, which generates an Interpretation Alignment matrix, showing the degree to which the interpretations align with the concepts. The domain expert can then use this information to further refine and/or improve the produced interpretations, and consequently the domain knowledge, by better understanding the extracted concepts.

2.1 Concept Extraction

Existing work on Concept Extractors (CEs) typically represents a concept as a set of images/superpixels (in case of computer vision), or as set of graphs in case of GNNs. More formally, we can

[1] Code is publicly available at https://github.com/dmitrykazhdan/gci

therefore represent a concept extraction approach f_{ext} as a function $f : (\mathcal{G}, \mathcal{M}) \rightarrow \mathcal{P}$, taking in a set of graphs $G \in \mathcal{G}$, together with a GNN model $M \in \mathcal{M}$ trained on this graph set, and returning a set of sets of graphs (i.e., the subset of the powerset of $\mathcal{G}$, defined as $\mathcal{P}$). Hence, $f_{ext}(G, M) = \{G_1, ..., G_m\}$ maps a given graph set G (the model's *training dataset*) and model M to a set of sets of graphs $\{G_1, ..., G_m\}$, where each set G_i represents a concept C_i.

2.2 Concept Interpretations

Once a CE has returned a set of concepts, the next step involves *interpreting* the extracted concepts, and attempting to describe what they may represent (e.g., 'I think this concept represents graphs with blue nodes'). More formally, this process may be seen as defining a set of *interpretation functions* $h : \mathcal{Z} \rightarrow \{0, 1\}$. Here, $h(g)$ returns *true* or *false*, depending on whether the given anticipated property is or is not present in a given graph g (e.g., 'this is a graph with blue nodes').

2.3 Interpretation-Concept Alignment

Intuitively, a 'good' interpretation for a concept is one which generally holds for all or most examples of that concept. For example, an interpretation of 'this concept represents graphs with blue nodes' can be considered good, if graphs of this concept are indeed predominantly graphs with blue nodes. Using the above definitions, we can measure the *alignment* between a given concept and an interpretation as $q : \mathcal{G} \times \mathcal{H} \rightarrow \mathbb{R}$, where $q(G, h)$ is defined as:

$$q(G, h) = \frac{|\{h(g) \text{ is true, } g \in G\}|}{|G|} \tag{1}$$

Hence, the alignment between concept samples G and an interpretation h is the fraction of the samples for which h is true, that is, the *precision* of h with respect to G. Consequently, given a set of concept samples $C = \{G_1, ..., G_m\}$, and a set of interpretations $H = \{h_1, ..., h_n\}$, we define an *Interpretation-Alignment* (IA) matrix as:

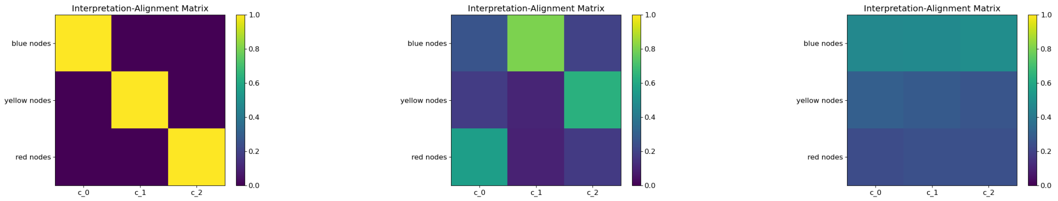
$$IA(C, H)_{i,j} = q(G_i, h_j), \forall i, j. \tag{2}$$

2.4 Further Interpretation Metrics

The above definition for interpretations lends itself to numerous further metrics that can be built on top of interpretations. For instance, we can introduce the notion of 'interpretation hierarchies', by seeing if one interpretation is a 'subset' of another, that is, $h_1 \subset h_2 \Leftrightarrow \forall g \in \mathcal{Z} : h_1(g) \Rightarrow h_2(g)$. Alternatively, we can also use this definiton to quantitatively explore relationships between interpretations by measuring their mutual information. We focus on the precision-based IA matrix in this work, leaving extensions of these metrics for future work.

Interpretation Completeness & Predictability In this work, we also build on top of the *concept completeness* metric introduced in (Yeh et al., 2020), and *concept predictability* metric introduced in Kazhdan et al. (2020b). The completeness of a set of concepts is defined as the accuracy of a classifier with respect to the end task, which uses *only* concept information as input (i.e., 'how predictable are the end-task labels from the concept information alone?') (Yeh et al., 2020). The predictability of a given concept from a given model is defined as the accuracy of a classifier trained to predict the values of that concept from the model's hidden space (further details are given in Kazhdan et al. (2020b)).

We build on top of the above notions, by observing that we can apply our interpretation functions *directly* to a model's training graph dataset, and thus obtain an *interpretation representation* of this dataset. We can then measure the completeness and predictability of this representation, in the same way as the completeness and predictability of a concept representation. We refer to these metrics as *interpretation completeness* and *interpretation predictability* in the remainder of this work. Intuitively, these metrics represent how *predictable* human-provided interpretations are from a given GNN (which serves as a proxy for how much the model learned this information), as well as how *complete* these interpretations are with respect to an end task (i.e. how well do these interpretations



(a) GCExplainer

(b) Noisy GCExplainer

(c) Random Explainer

Figure 2: IA-Matrices for the different heuristics. Each column corresponds to an extracted concept (c_0-c_2), and each row corresponds to a defined interpretation (h_0-h_2).

describe the actual task). Collectively, these metrics demonstrate how *relevant* interpretations are for the end task Kazhdan et al. (2020b).

3 Experiments & Results

In this section, we demonstrate the utility of the GCI framework on a range of synthetic baselines, as well as on a practical use-case of molecular property prediction. Similarly to Ying et al. (2019), we rely on synthetically-generated Barabási-Albert graphs for our synthetic baseline experiments.

3.1 Concept Extractor Evaluation

Firstly, we demonstrate how GCI can be used to quantitatively evaluate different CEs of varying quality, and compare it with a visual inspection approach. In this setup, we relied on a synthetic graph classification dataset with a trivial concept structure, and simulated a reduction in CE quality by injecting random perturbations into its output.

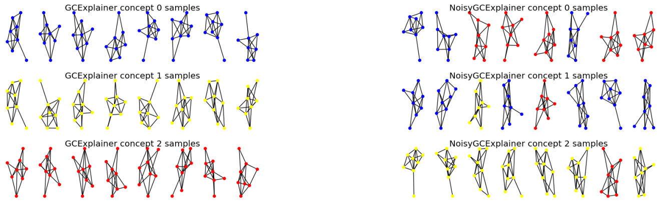
Dataset: We generated 1000 Barabási-Albert graphs. 50% of these graphs were coloured blue, 25% were coloured red, and 25% were coloured yellow. The blue graphs were assigned class label 0, whilst the red and yellow graphs were assigned class label 1. Overall, this dataset represents a graph classification task consisting of 2 classes, in which class 1 is composed of 2 distinct concepts (yellow and red node colours).

GNN Model: We trained a standard Graph Convolutional Network (Zhang et al., 2019) on the above binary graph classification task, achieving 100% accuracy.

Concept Extractor Baselines: We used 3 different baselines for our CEs: *GCExplainer*, *NoisyGCExplainer*, and *RandomExplainer*. GCExplainer uses the methodology described in Magister et al. (2021). NoisyGCExplainer works by taking the output of a GCExplainer, and randomly moving a fraction θ of the subgraphs from one concept to another (for this experiment, we set $\theta = 20\%$). This baseline represents a more noisy, imperfect CE. Finally, RandomExplainer takes the output of a GCExplainer, and randomly shuffles the subgraphs across the concepts, representing a baseline random concept predictor. We assumed the *number* of concepts is known in these experiments, and set that to 3 for the GCExplainers. Overall, these baselines represent CEs of varying quality, in which a reduction in quality is achieved by injecting randomness into concept outputs. Further details are given in Appendix B.1.

Interpretations: We provided 3 *ground truth* interpretations, defined as $h_0(g) \Leftrightarrow$ 'g is a graph with red nodes', $h_1(g) \Leftrightarrow$ 'g is a graph with blue nodes', $h_2(g) \Leftrightarrow$ 'g is a graph with yellow nodes'.

Results The IA matrices between the different explainer concepts and interpretations are presented in Figure 2. It is clear that GCExplainer has near-perfect alignment with the interpretations, whilst the other two explainers demonstrate a significantly lower alignment. Extracted concept samples are shown in Figure 3. Crucially, the IA matrix quantitatively shows the vast difference between explainers (with GCExplainer aligning the best). Such a conclusion would have been much more difficult to draw from simple visual inspection.


GCExplainer concept 0 samples
GCExplainer concept 1 samples
GCExplainer concept 2 samples
NoisyGCExplainer concept 0 samples
NoisyGCExplainer concept 1 samples
NoisyGCExplainer concept 2 samples
RandomExplainer concept 0 samples
RandomExplainer concept 1 samples
RandomExplainer concept 2 samples

(a) GCExplainer (b) Noisy GCExplainer (c) Random Explainer

Figure 3: Representative concept samples for the different explainers. Notice that each row depicts the examples contained within a single concept. The more similar the graphs within the concept, the higher the purity of the concept; hence, the explainer is better. Observe how this information is accurately summarised in the IA matrix in Figure 2.

3.2 Intepretation Alignment Grading

In this section, we demonstrate how GCI can be used to quantitiatively measure the degree of alignment between human interpretations and extracted concepts, using a synthetic graph classification task. In particular, we rely on CEs of varying *granularity* (with one CE extracting relatively more lower-level concepts), and show how GCI can be used to determine which sets of interpretations best describe which extractor. As before, we compare this to a visual inspection approach.

Dataset: We generated 1000 Barabási-Albert graphs synthetically. 50% of these graphs were coloured blue. The remaining 50% of the graphs were coloured red. Furthermore, we attached a square subgraph structure to 80% *of the red graphs*. The blue graphs were assigned class label 0, and the red graphs a class label 1. Overall, this dataset represents a graph classification task consisting of 2 classes, in which one of the classes has 2 highly correlated concepts (a square subgraph, and the colour red).

GNN Model: We trained a standard GCN on the above binary graph classification task, achieving 100% accuracy.

Concept Extractor Baselines: We used 2 different GCExplainer baselines: one with the number of concepts set to 2, and another one set to 3. These two baselines represent CEs of varying *granularity*.

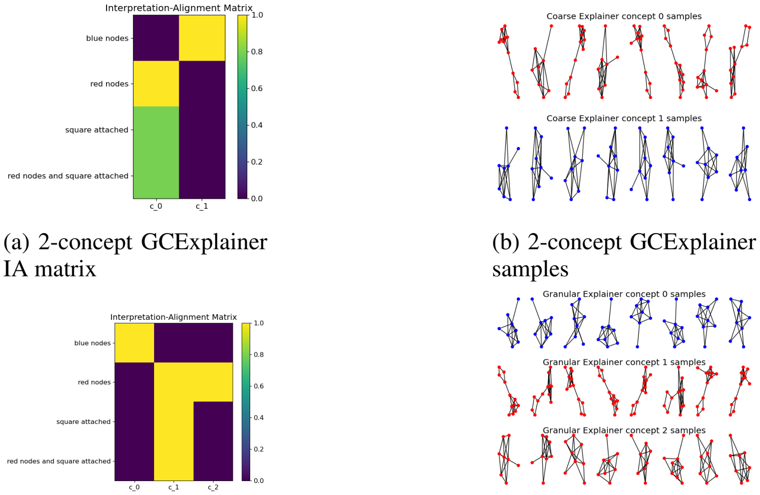
Interpretations: We provided 4 different interpretations defined as follows: $h_0(g) \Leftrightarrow$ 'g is a blue graph'; $h_1(g) \Leftrightarrow$ 'g is a red graph'; $h_2(g) \Leftrightarrow$ 'g is a graph with an attached square subgraph'; $h_3(g) \Leftrightarrow h_1(g)$ AND $h_2(g)$.

Results: Figure 4 shows the IA matrices and extracted concept samples for both explainers. Importantly, we observe that at lower granularity, only h_0 and h_1 accurately describe the extracted concepts, whilst at higher granularity, the CE was successfully able to separate out the two concepts of class 1, such that all 4 heuristics accurately describe the different concepts. Such a conclusion regarding which sets of interpretations better describe the extracted concepts would have been very difficult to draw simply by observing the concept samples (shown in Figure 4 (b) & (d)).

3.3 Model-Interpretation Completeness & Predictability

In this section, we demonstrate how GCI can be used to measure the completeness of a set of interpretations with respect to an end task, and the predictability of these interpretations from a given model (as discussed in Section 2). Furthermore, we demonstrate that GCI can be used to verify concepts, even if they are *not* relevant to an end-task.

Dataset: We generated 1000 Barabasi-Albert graphs synthetically. 50% of these graphs were coloured blue and assigned a class label 0, whilst the other 50% were coloured red and assigned a class label 1. For each of the classes, we randomly selected 15% of the samples, and coloured one randomly selected node in *each* graph purple. We then randomly selected 15% of the samples from *each* class again, and added a square subgraph to them. Overall, this dataset represents a graph classification task where there are 2 concepts directly related to the end task (the red and blue colour), and 2 concepts that are *independent* of the end task (the purple node and the square structure).



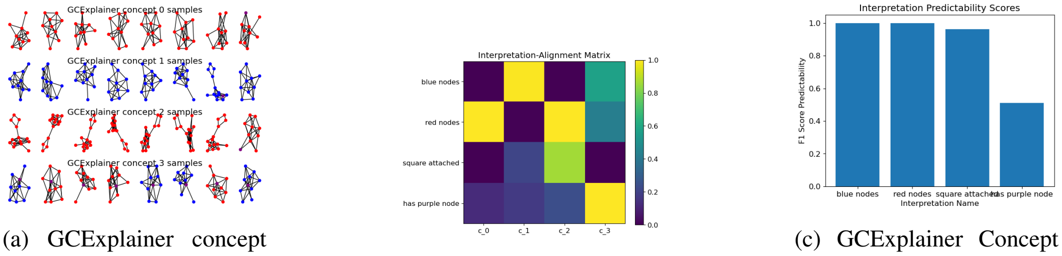
(a) 2-concept GCExplainer IA matrix

(b) 2-concept GCExplainer samples

(c) 3-concept GCExplainer IA matrix

(d) 3-concept GCExplainer samples

Figure 4: Representative concept samples (right) and corresponding IA matrices (left) for the concepts of two different explainers of varying granularity based on the number of concepts they can represent (a) & (b) vs (c) & (d), for 2 and 3 concepts, respectively.



(a) GCExplainer concept samples

(b) GCExplainer IA Matrix

(c) GCExplainer Concept Predictability

Figure 5: Representative concept samples (a), corresponding IA matrices (b) for the concepts, and (c) GCExplainer Concept Predictability Scores. Notice that concepts relevant to the task have higher F1 Score predictability than irrelevant concepts.

GNN Model: We trained a standard GCN on the above binary graph classification task, achieving 99% accuracy.

Concept Extractor Baseline: We use a GCExplainer with the number of concepts set to 4 in this example (as before, we assumed the number of concepts is known in these experiments).

Interpretations: We provided 4 different *ground truth* interpretations, corresponding to the 4 underlying concepts: $h_0(g) \Leftrightarrow$ 'g is a blue graph'; $h_1(g) \Leftrightarrow$ 'g is a red graph'; $h_2(g) \Leftrightarrow$ 'g is a graph with an attached square subgraph'; $h_3(g) \Leftrightarrow$ 'g contains a purple node'.

Results: Firstly, we show the concept samples from the CE in Figure 5a, as well as the IA matrix in Figure 5b. Importantly, *all* underlying concepts have been picked up by the CE successfully (with above 70% precision).

Next, we applied our interpretation functions to the input graph dataset to obtain its interpretation representation, and computed the completeness of that representation (as discussed in Section 2). We obtained a completeness score of 99% for this interpretation representation. Furthermore, we measured the predictability of this representation, with the results shown in Figure 5c (further details are given in Appendix B.2). Overall h_2 and h_3 obtained relatively lower predictability scores.

The above results demonstrate that we can use GCI to measure predictability and completeness of interpretations in the same way completeness and predictability is measured for concepts. Furthermore, the above results are consistent with the findings made in Kazhdan et al. (2020c), which showed that concepts with a lower dependence on the end-task will usually have lower predictability, since the model does not need to retain information about these concepts in order to achieve high performance.

Importantly, Figure 5b also demonstrates that GCI and GCExplainer successfully extracted and verified *all* underlying concepts, not just the ones directly related to the end task. This is significant, since this indicates that GCI and corresponding concept extractors are not restricted to only extracting and verifying concepts directly related to the end task.

Overall, the above results show how GCI can be used to (i) measure how complete a given set of interpretations is with respect to an end task, (ii) measure the degree to which a given model learns these interpretations (via predictability), (iii) verify extracted concepts using these interpretations, including concepts *not* related to the end-task

3.4 Application: Molecular Property Prediction

In this section, we demonstrate how GCI can be applied to a practical use-case of *molecular property prediction* Walters & Barzilay (2020). This is achieved by integrating GCI with *TorchDrug*[2] (a framework for drug discovery with machine learning), and using one of the available molecular prediction benchmarks from there Zhu et al. (2022); Wu et al. (2018).

When dealing with chemistry tasks such as molecular property prediction, there are multiple ways of representing information about the underlying molecular graphs, at varying levels of granularity. For instance, a molecule can be described by its molecular mass (high-level information), by its SMILES representation Weininger (1990), or by its molecule type, etc.

Intuitively, selected interpretations used to describe extracted concepts should ‘match’ the granularity of the extracted concepts in order to describe them well. It was previously shown that CEs, such as GCExplainer Magister et al. (2021), typically group graphs sharing similar subgraph structures. Hence, we need to rely on interpretations, that also operate on the subgraph level when describing graphs.

Consequently, we rely on chemical *functional groups* as a potential source of interpretations in this work. Informally, a functional group is defined as a group of atoms responsible for the characteristic reactions of a particular compound Carey & Sundberg (2007). Hence, functional groups represent graph information at a ‘subgraph level’, and are a suitable choice for representing interpretations. We leave the exploration of other types of interpretations for future work.

3.4.1 Setup

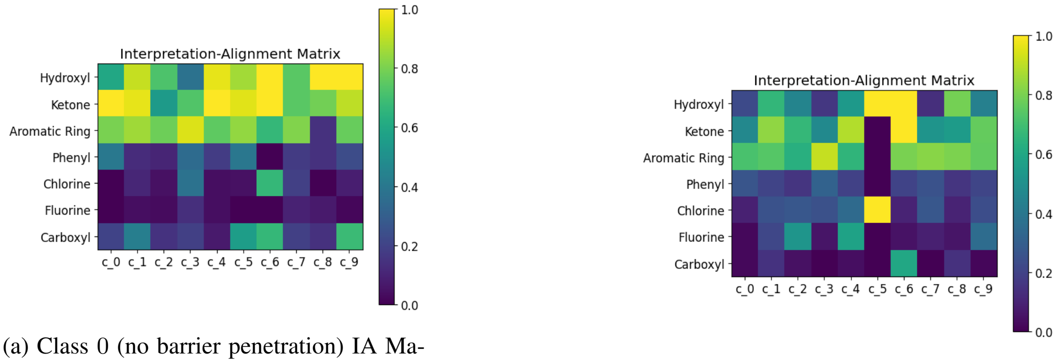
Dataset: We used the Blood-Brain Barrier Penetration (BBBP) dataset as our benchmark Martins et al. (2012). This benchmark consists of a binary graph classification task, where the goal is to predict whether a given molecule will penetrate the blood-brain barrier. The blood-brain barrier blocks most drugs, hormones and neurotransmitters. Hence, the penetration of the barrier forms a long-standing issue in development of drugs targeting the central nervous system Wu et al. (2018); Martins et al. (2012).

GNN Model: We trained a standard GCN, achieving 0.88 AUROC performance (consistent with state-of-the-art findings). Further details can be found in Appendix B.3.

Concept Extractor Baseline: We used the GCExplainer CE with different configurations for the number of concepts. We show the results of the best-performing configuration here, and include others in Appendix C.4.

Interpretations: In this work, we implemented 6 well-known functional groups, including: *hydroxyl group*, *ketone group*, *phenyl group*, *chlorine group*, *fluorine group*, *carboxyl group*. Finally, we also implemented an interpretation which checks whether a molecule has at least one *aromatic ring* Sandler & Karo (2013). This set of interpretations represents functional groups that are rela-

[2] https://torchdrug.ai/



(a) Class 0 (no barrier penetration) IA Matrix.

(b) Class 1 (barrier penetration) IA Matrix.

Figure 6: IA matrices for the two Blood-Brain Barrier Penetration (BBBP) classes, using GCExplainer as the concept extractor, and functional groups as interpretations.

tively conceptually simple. An overview of these interpretations is given in Appendix C.3. We leave implementation of more functional groups and other forms of chemistry heuristics for future work.

3.4.2 Results

Firstly, we analyse concepts extracted by GCExplainer from samples of each of the classes, and show the corresponding class IA matrices in Figure 6.

Importantly, Figure 6 demonstrates how GCI can be used to (i) show which interpretations are important/relevant for an end-task, and (ii) how these interpretations differ accross classes. In particular, we see that the top 3 interpretations (hydroxyl, ketone, and the aromatic ring) generated a large alignment with some of the extracted concepts for both classes.

Furthermore, we can use the IA matrices to see which interpretations contribute to *individual classes*. For instance, the Chlorine functional group does not resonate with any concept from class 0, but strongly resonates with the 6th concept (c_5) in class 1, together with hydroxyl. In practice, such an analysis can be a useful way to quickly visualise and interpret the key concepts contributing to individual classes.

Similarly to the previous experiment, we also measured the completeness of the implemented interpretations, achieving a 0.76 AUCROC score, only 0.12 points below the original model score of 0.88.

Collectively, these results demonstrate how GCI can be used to (i) encode domain knowledge heuristics (in this case - functional groups) as interpretations, (ii) use these interpretations to see which extracted concepts are aligned with them the most, (iii) verify how *complete* these interpretations are, with respect to the end task. In the above experiment, the implemented functional groups already achieved a completeness score close to that of the original model, indicating that these interpretations serve as a good representation of the end task. We leave implementation of more types of interpretations representing the chemistry domain for future work. Further results can be found in Appendix C.

4 Conclusions

We introduced GCI: a Graph Concept Interpretation framework, used to quantitatively interpret concepts extracted from GNNs. Using several case-studies, we demonstrate how GCI can be used to (i) quantitatively evaluate different concept extractors, (ii) quantitatively measure alignment between a set of concepts & human interpretations, (iii) measure the task completeness of human interpretations. Furthermore, we show how GCI can be applied to a molecular property prediction case-study, and used to interpret the extracted concepts. Given the rapidly-increasing interest in concept-based explanations of GNN models, we believe GCI can play an important role in mining new concepts from GNNs trained on other drug discovery tasks.

ACKNOWLEDGMENTS

DK acknowledges support from the EPSRC ICASE scholarship and GSK.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pp. 9505–9515, 2018.

Francis A Carey and Richard J Sundberg. *Advanced organic chemistry: part A: structure and mechanisms*. Springer Science & Business Media, 2007.

Botty Dimanov. *Interpretable Deep Learning: Beyond Feature-Importance with Concept-based Explanations*. PhD thesis, University of Cambridge, 2021.

Botty Dimanov, Umang Bhatt, Mateja Jamnik, and Adrian Weller. You shouldn't trust me: Learning models which conceal unfairness from multiple explanation methods. In *European Conference on Artificial Intelligence*, 2020.

Bettina Finzel, Anna Saranti, Alessa Angerschmid, David Tafler, Bastian Pfeifer, and Andreas Holzinger. Generating explanations for conceptual validation of graph neural networks: An investigation of symbolic predicates learned on relevance-ranked sub-graphs. *KI-Künstliche Intelligenz*, pp. 1–15, 2022.

Dobrik Georgiev, Pietro Barbiero, Dmitry Kazhdan, Petar Veličković, and Pietro Liò. Algorithmic concept-based explainable reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6685–6693, 2022.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *AAAI*, 2019a.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32, 2019b.

Dmitry Kazhdan, Botty Dimanov, Mateja Jamnik, and Pietro Liò. Meme: generating rnn model explanations via model extraction. *arXiv preprint arXiv:2012.06954*, 2020a.

Dmitry Kazhdan, Botty Dimanov, Mateja Jamnik, Pietro Liò, and Adrian Weller. Now you see me (CME): concept-based model extraction. In Stefan Conrad and Ilaria Tiddi (eds.), *Proceedings of the CIKM 2020 Workshops co-located with 29th ACM International Conference on Information and Knowledge Management (CIKM 2020), Galway, Ireland, October 19-23, 2020*, volume 2699 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020b. URL `http://ceur-ws.org/Vol-2699/paper02.pdf`.

Dmitry Kazhdan, Botty Dimanov, Mateja Jamnik, Pietro Liò, and Adrian Weller. Now you see me (cme): concept-based model extraction. *arXiv preprint arXiv:2010.13233*, 2020c.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2673–2682. PMLR, 2018. URL `http://proceedings.mlr.press/v80/kim18d.html`.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 267–280. Springer, 2019.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Lucie Charlotte Magister, Dmitry Kazhdan, Vikash Singh, and Pietro Liò. Gcexplainer: Human-in-the-loop concept-based explanations for graph neural networks. *arXiv preprint arXiv:2107.11889*, 2021.

Lucie Charlotte Magister, Pietro Barbiero, Dmitry Kazhdan, Federico Siciliano, Gabriele Ciravegna, Fabrizio Silvestri, Mateja Jamnik, and Pietro Lio. Encoding concepts in graph neural networks. *arXiv preprint arXiv:2207.13586*, 2022.

Ines Filipa Martins, Ana L Teixeira, Luis Pinheiro, and Andre O Falcao. A bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling*, 52(6):1686–1697, 2012.

Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10772–10781, 2019.

Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*, 2018.

Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 2021. doi:10.23915/distill.00033. https://distill.pub/2021/gnn-intro.

Stanley R Sandler and Wolf Karo. Organic functional group preparations. 2013.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

W Patrick Walters and Regina Barzilay. Applications of deep learning in molecule generation and molecular property prediction. *Accounts of chemical research*, 54(2):263–270, 2020.

David Weininger. Smiles. 3. depict. graphical depiction of chemical structures. *Journal of chemical information and computer sciences*, 30(3):237–243, 1990.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Han Xuanyuan, Pietro Barbiero, Dobrik Georgiev, Lucie Charlotte Magister, and Pietro Lió. Global concept-based interpretability for graph neural networks via neuron analysis. *arXiv preprint arXiv:2208.10609*, 2022.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33:20554–20565, 2020.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, et al. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems*, a.

Mateo Espinosa Zarlenga, Pietro Barbiero, Zohreh Shams, Dmitry Kazhdan, Umang Bhatt, and Mateja Jamnik. On the quality assurance of concept-based representations. b.

Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.

Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 119–134, 2018.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Zhaocheng Zhu, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, et al. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.

A Related Work

Concept Extraction Existing concept extraction approaches have primarily been studied in the field of Computer Vision. These approaches extract concepts from a CNNs hidden space in an *unsupervised* fashion, representing these concepts as groups of superpixels (Kazhdan et al., 2020b; Dimanov, 2021; Ghorbani et al., 2019b; Kim et al., 2017; Yeh et al., 2020; Zarlenga et al., a). More recently, variants of these approaches have been applied to GNNs as well, where concepts are extracted in the form of subgraphs in an unsupervised fashion (Magister et al., 2021; 2022). In this work, we rely on GCExplainer (Magister et al., 2021) as the primary benchmark in our experiments.

Concept-based Explanations & Graph Neural Networks Concept-based explanations have been applied to GNNs in two primary forms: for performing concept extraction from GNN models (Magister et al., 2021; Xuanyuan et al., 2022), and for using concepts to build GNN models *interpretable by design* (Magister et al., 2022; Georgiev et al., 2022). Importantly, similarly to computer vision, in all these cases the concepts are either assumed to be known beforehand, or are interpreted via visual inspection. Our GCI framework builds on top of these approaches by allowing quantitative interpretation of potentially-novel concepts.

Concept Verification Several methods have been proposed for evaluating concept extraction approaches, besides using visual inspection. Work by Ghorbani et al. (2019b) and Yeh et al. (2020) relies on human studies to evaluate properties such as *concept coherence*. Work by Magister et al. (2021) introduces *concept purity* for measuring the difference between graphs of a given concept, which relies on computing graph edit distance. Work in Zarlenga et al. (b) propose mutual information based metrics to evaluate quality of concept representations. Finally, work by Finzel et al. (2022) introduces a validation framework for GNN explainers, in which a set of Inductive Logic Programming rules are produced from domain experts and GNN explanations. These are then compared with each other for validation. Importantly, these approaches either rely on time-intensive user studies, or metrics which are computationally-infeasible for a large number of graphs (such as edit distance), or assume the concepts have already been interpreted and labelled (as in Zarlenga et al. (b)), or are applicable to niche types of graph datasets (as in the last case). In contrast, GCI is fast to setup, computationally feasible, is used for *intepreting* concepts, and is applicable to any type of graph data.

B Experimental Setup

B.1 Randomised GCExplainers

As discussed in Magister et al. (2021), GCExplainer works by clustering a given dataset in the activation space of a given GNN, with the resulting clusters then representing the underlying concepts. In this work we use the setup from Magister et al. (2021), where we rely on K-means clustering as the clustering algorithm. In this case, every graph sample is assigned to one cluster in the hidden space, and is therefore part of exactly one concept (i.e., the concept representation is effectively one-hot encoded).

Consequently, our randomised versions of GCExplainer work by randomly reassigning graph samples from their original cluster to a random one, thereby representing random concept assignment.

B.2 Concept Predictability Measurement

For measuring predictability of concepts, we rely on the setup from Kazhdan et al. (2020b). In particular, for a given concept and GNN model, we train a Logistic Regression classifier to predict the concept values from the activations of the last layer of the GNN (using other layers resulted in worse performance).

B.3 GNN Architectures

Synthetic Baselines For all synthetic baselines, we rely on a Graph Convolutional Neural Network (GCN) Kipf & Welling (2016), with 3 convolutional layers, 1 linear layer, and global mean pooling. Further details can be found in the codebase.

Molecular Property Prediction For the BBBP task, we relied on a GCN with 4 convolutional layers, 1 linear layer, and a and global mean pooling layer.

C BBBP Further Results

C.1 IA-Matrix for all samples

Figure 7 shows the IA matrix for both classes (i.e., for all dataset samples).

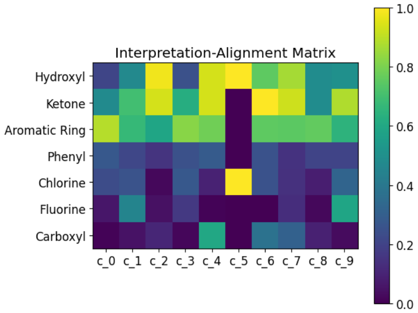


Figure 7: IA matrix computed from all samples of the BBBP dataset

C.2 Further Concept Samples

Figure 8, Figure 9, and Figure 10 show the concept samples of our GCExplainer for each of the 10 extracted concepts, for Class 1, Class 0, and all samples, respectively.

Overall, the IA Matrices shown in Figure 6 serve as a compact representation of the molecules in each concept.

For instance - notice that c_6 of Class 0 was in fact an outlier molecule extracted by GCExplainer, which indeed was predominantly consisting of *Chlorine* and *Hydroxyl* molecules, as demonstrated in the IA Matrix.

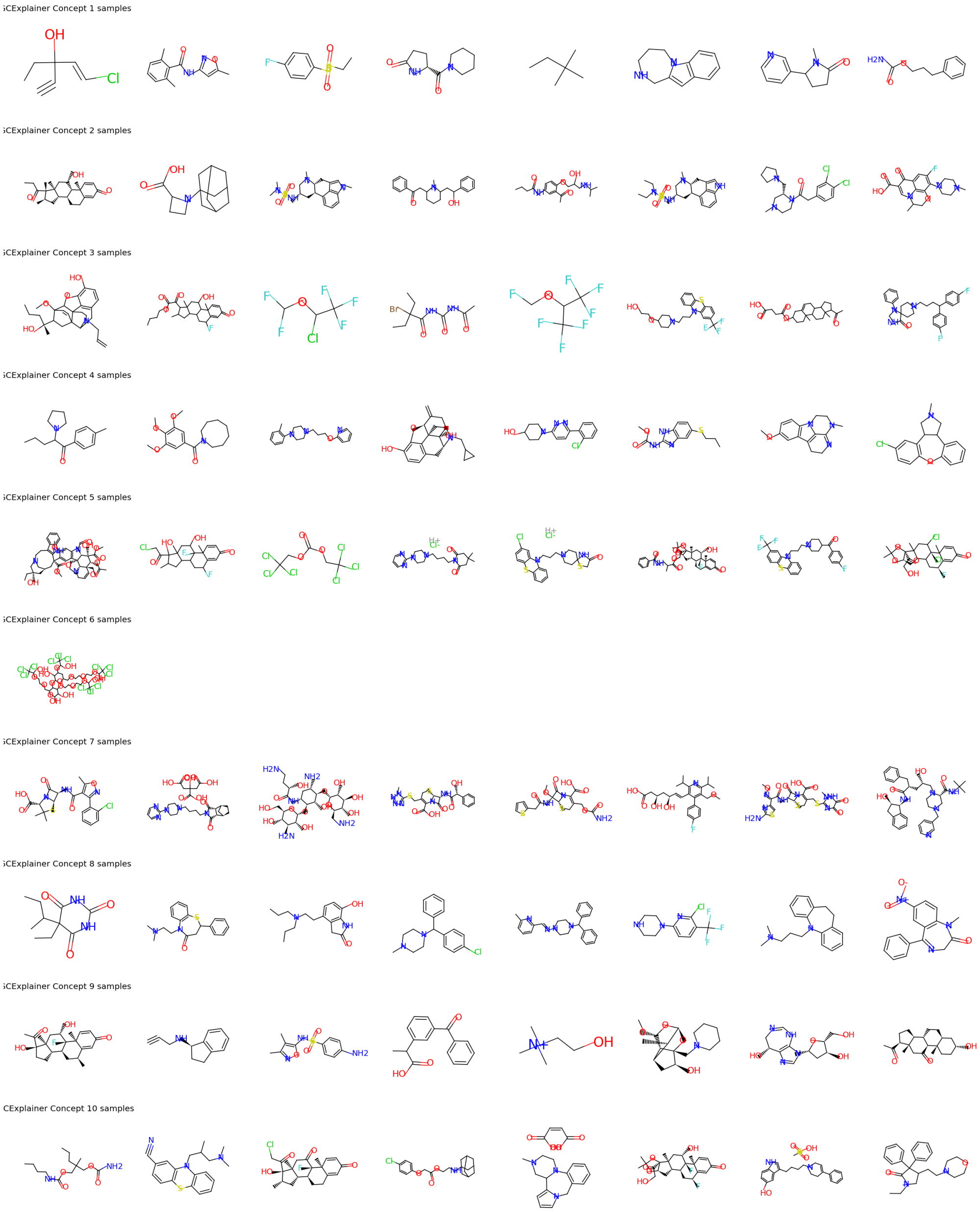

CExplainer Concept 1 samples
CExplainer Concept 2 samples
CExplainer Concept 3 samples
CExplainer Concept 4 samples
CExplainer Concept 5 samples
CExplainer Concept 6 samples
CExplainer Concept 7 samples
CExplainer Concept 8 samples
CExplainer Concept 9 samples
CExplainer Concept 10 samples

Figure 8: GCExplainer concept samples for the BBBP task, Class 1

C.3 Interpretations

Figure 11 shows a few interpretation samples for each of our 7 heuristics.

C.4 Varying the number of concepts

Below, we plot IA-Matrices for a varied number of GCExplainer concepts in the BBBP task, for Class 1.

Importantly, GCI can be used to quickly explore and compare concept representations at varying hierarchies.

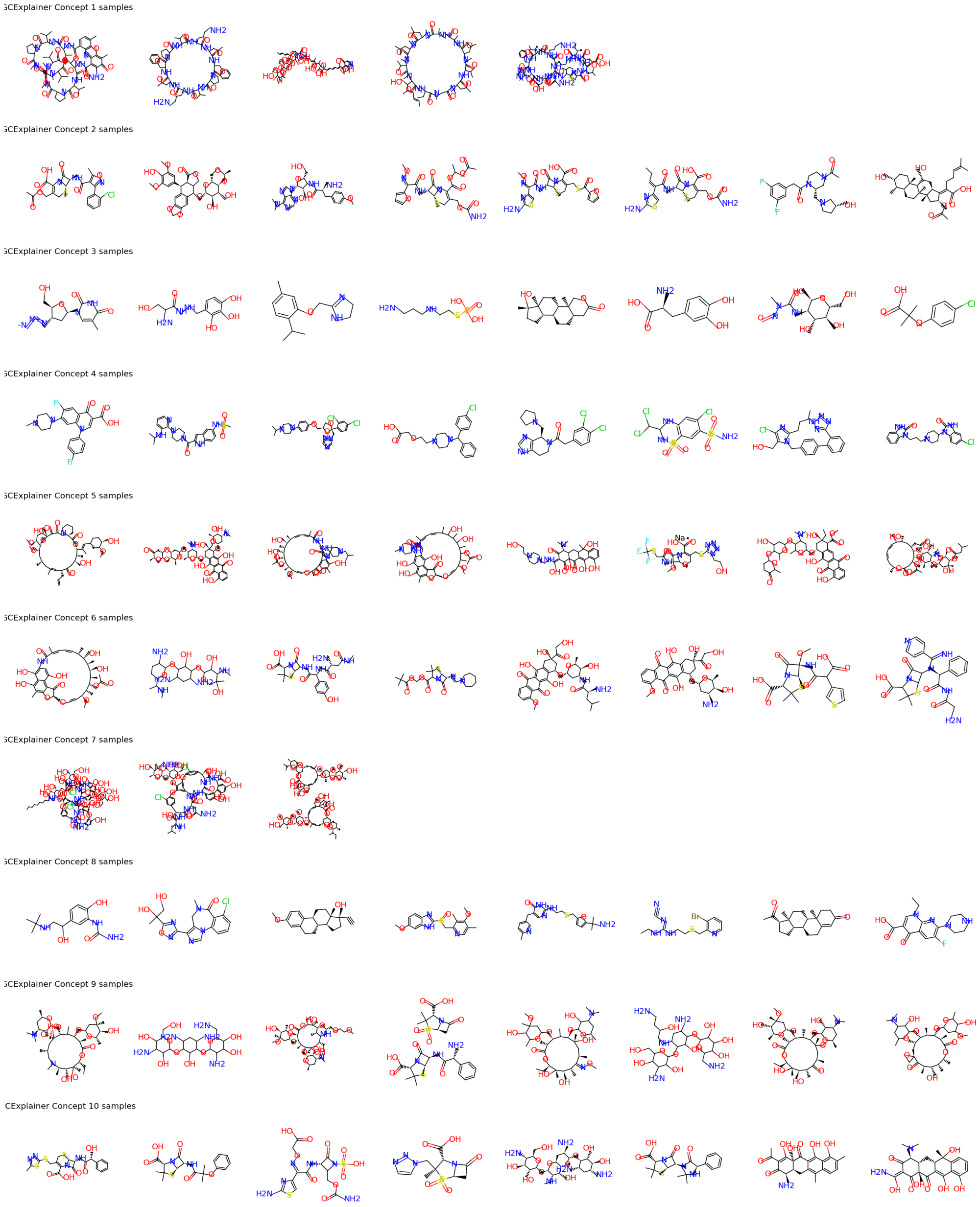

GCExplainer Concept 1 samples
GCExplainer Concept 2 samples
GCExplainer Concept 3 samples
GCExplainer Concept 4 samples
GCExplainer Concept 5 samples
GCExplainer Concept 6 samples
GCExplainer Concept 7 samples
GCExplainer Concept 8 samples
GCExplainer Concept 9 samples
CExplainer Concept 10 samples

Figure 9: GCExplainer concept samples for the BBBP task, Class 0

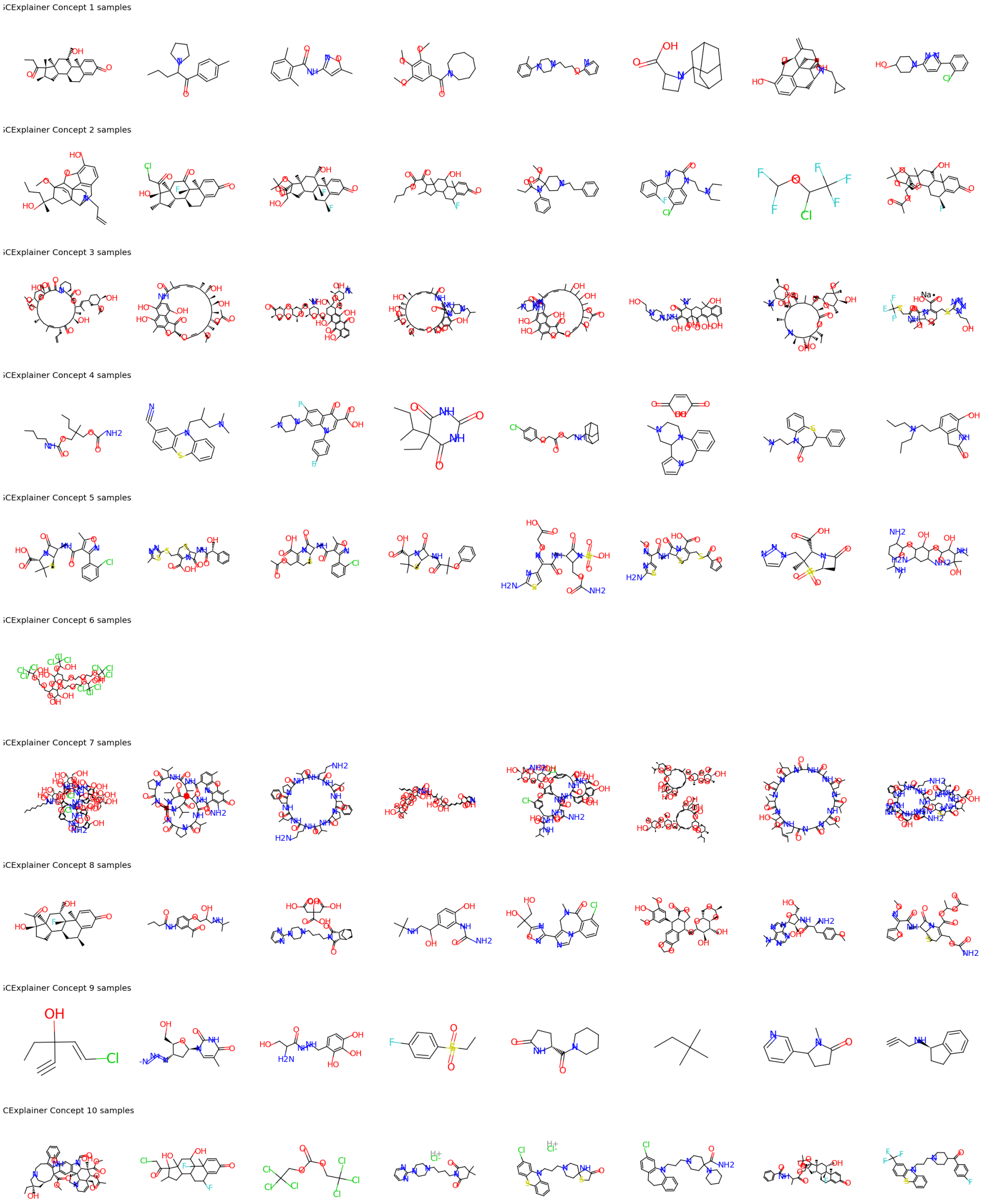

iCExplainer Concept 1 samples
iCExplainer Concept 2 samples
iCExplainer Concept 3 samples
iCExplainer Concept 4 samples
iCExplainer Concept 5 samples
iCExplainer Concept 6 samples
iCExplainer Concept 7 samples
iCExplainer Concept 8 samples
iCExplainer Concept 9 samples
CExplainer Concept 10 samples

Figure 10: GCExplainer concept samples for the BBBP task, all samples

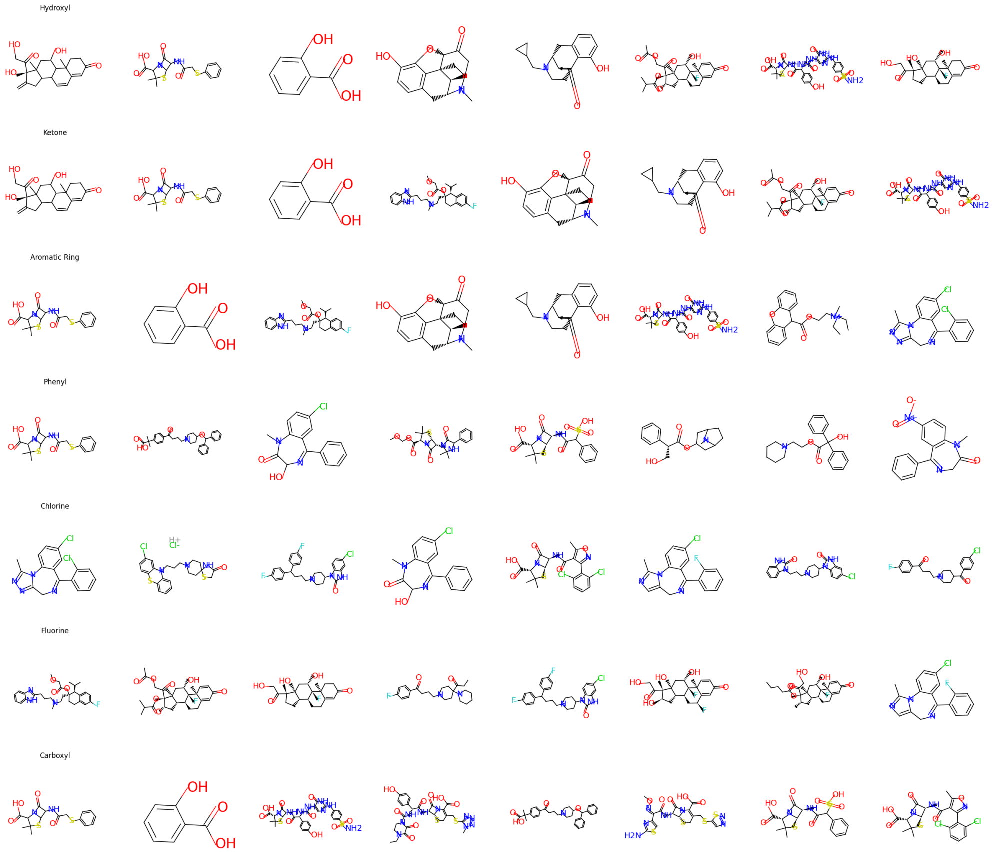


Figure 11: Graph samples for each of our interpretations used in the BBBP task

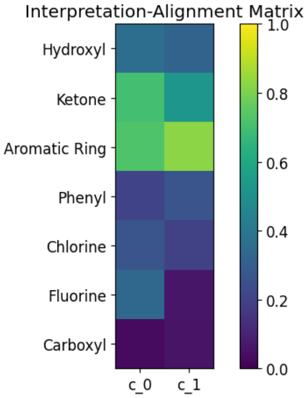


Figure 12: IA-Matrix for Class 1, using 2 concepts for GCExplainer

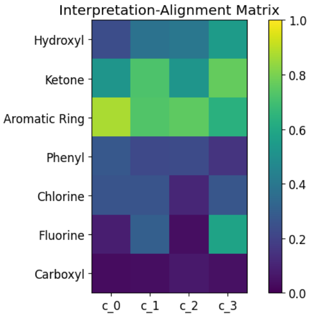


Figure 13: IA-Matrix for Class 1, using 4 concepts for GCExplainer

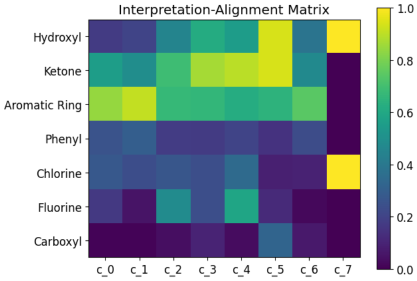


Figure 14: IA-Matrix for Class 1, using 8 concepts for GCExplainer

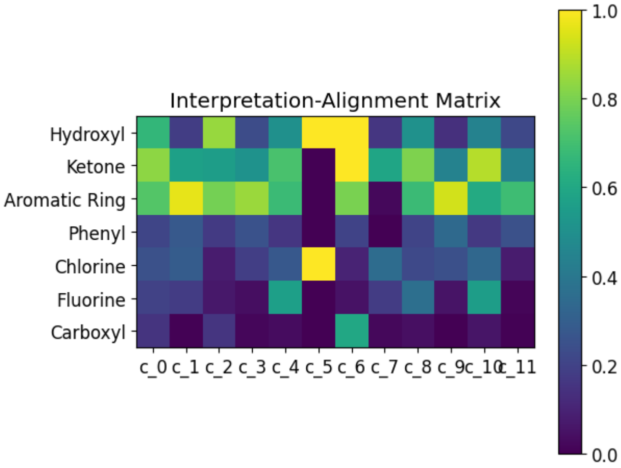


Figure 15: IA-Matrix for Class 1, using 12 concepts for GCExplainer

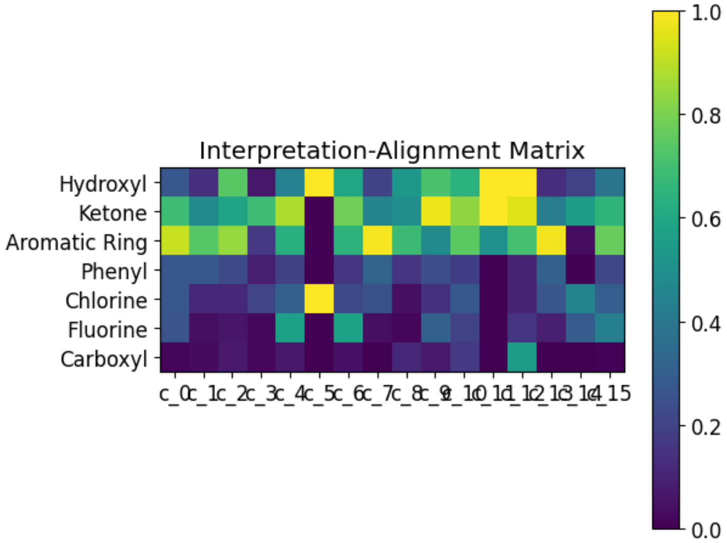


Figure 16: IA-Matrix for Class 1, using 16 concepts for GCExplainer

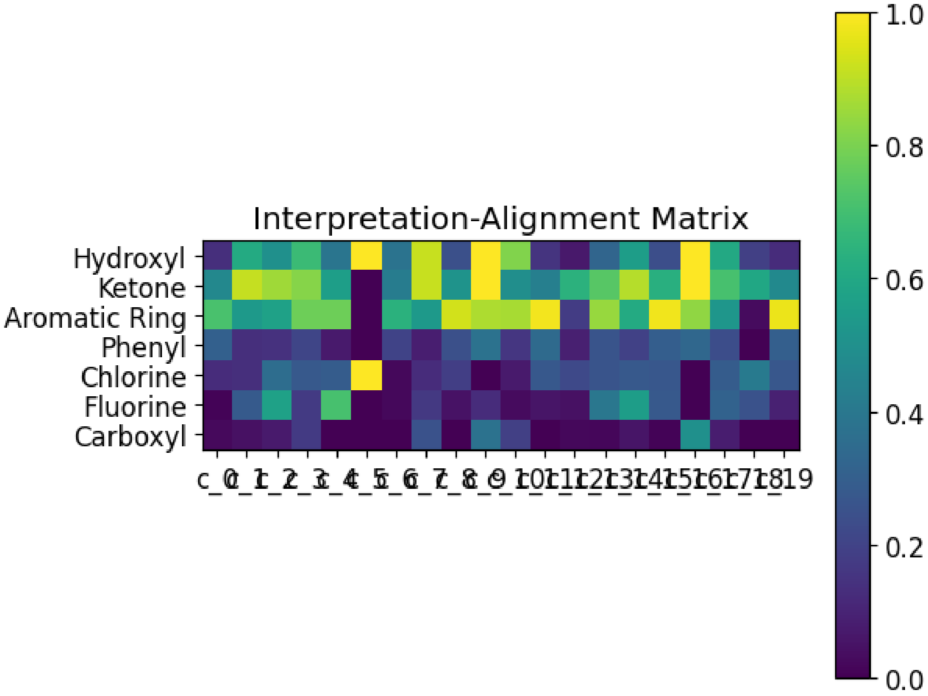


Figure 17: IA-Matrix for Class 1, using 20 concepts for GCExplainer

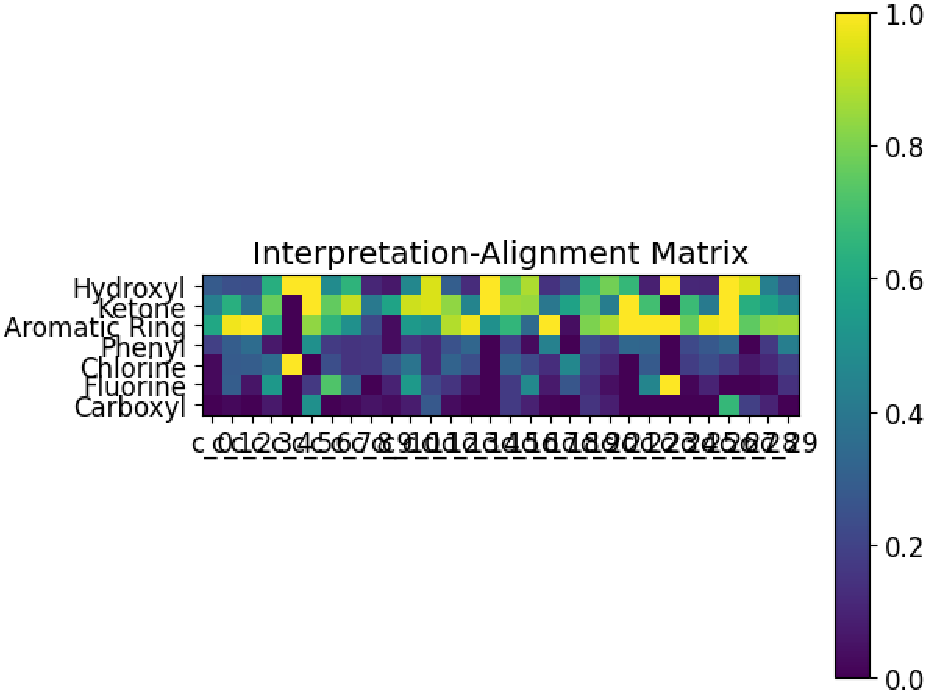


Figure 18: IA-Matrix for Class 1, using 30 concepts for GCExplainer