INTERPRETABLE PREDICTIONS FOR GRAPH CLASSIFICATION USING VARIATIONAL INFORMATION PURSUIT

Andrei Vlad Dome
McMaster University
Hamilton, ON L8S 4L8, Canada
andreivlad.dome@gmail.com

ABSTRACT

Graph Neural Networks (GNNs) have become ubiquitous for graph classification tasks. Although, their high accuracy comes with a sacrifice in interpretability, which is a pitfall in cases where transparency about a model's decision-making process is important, such as when employed for scientific discovery purposes or in high-risk scenarios. Post-hoc explainability methods can be used to explain a black-box model's internal reasoning, but may not always be reliable. As a result, models that are "interpretable-by-design" have emerged. One such method is Variational Information Pursuit (V-IP), which is a neural network-based method that sequentially asks user-interpretable, task-relevant queries about data until a prediction can be made with some sufficient level of confidence, resulting in a sequence of queries and answers that provide full transparency about the model's decision-making process. In this paper, we propose a framework for creating induced subgraph enumeration-based query sets for V-IP, in order to produce interpretable predictions for graph classification tasks. We demonstrate the efficacy of this framework by crafting a domain-specific query set for a graph classification task from chemistry, mutagen classification, and show that V-IP achieves test accuracies that beat those of black-box GNNs and another neural network-based interpretable-by-design model. Finally, we qualitatively show how V-IP's explanations provide valuable insight into how certain functional groups (specific subgraphs) of a molecule play a role in it being classified as mutagenic or not, paving the way for future domain-centric research in Explainable AI for mutagen classification and other graph-based tasks. [1]

1 INTRODUCTION

Many types of data can be represented as graph data structures, such as molecules, transportation networks, and financial transaction networks. Machine Learning (ML) techniques have been applied to solve graph-related tasks, from classifying molecules by chemical properties to fraud detection. Graph Neural Networks (GNNs) have become ubiquitous for such tasks due to their typically high accuracy and ability to learn complex patterns in graph data without the need for feature engineering. Although GNNs are black-box, and thus lack trustworthiness and transparency, which can be important when using them for the purpose of scientific discovery or in high-stake scenarios. As a result, methods have emerged that try to explain the internal reasoning behind the decisions of black-box neural networks, termed *post-hoc explainability*. Although, recent research has highlighted that these methods do not provide correctness guarantees and may fail to fully explain the model's decision-making process, leading to criticism about their lack of reliability (Kindermans et al., 2022; Adebayo et al., 2018; Rudin, 2019). Also, post-hoc methods typically rely on importance score-based explanations of raw features, which may lack the higher-level semantics necessary to produce useful explanations for some tasks (Chattopadhyay et al., 2023a). Consequently, the need for methods that are *interpretable-by-design* has gained popularity, i.e. methods where the model provides user-interpretable explanations that are inherently aligned with the model's internal reasoning process (Chattopadhyay et al., 2023a).

[1] Code is available at https://github.com/vladnotandrei/vip-interpretable-graph-classification

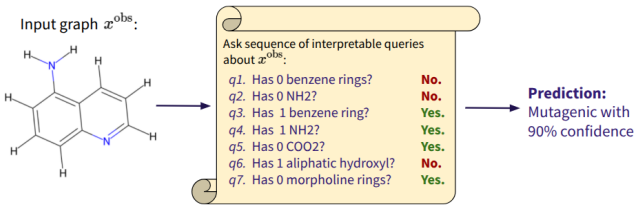


Figure 1: Illustration of interpretable-by-design framework on mutagen classification task. The query set consists of questions about functional groups (specific subgraphs) of a molecule. Given a molecule x^{obs}, a sequence of interpretable queries is composed, until a prediction about its class (mutagenic or non-mutagenic) can be made with a sufficient level of confidence.

One such interpretable-by-design method is Variational Information Pursuit (V-IP) (Chattopadhyay et al., 2023a). It uses two neural networks to approximate Information Pursuit (IP) (Geman & Jedynak, 1996) — a greedy algorithm for composing sequences of user-interpretable queries about data in order of information gain until a prediction can be made with some sufficient level of confidence (See Figure 1). Having been applied to images and text data in previous work (Chattopadhyay et al., 2023a), our work aims to provide a framework for applying V-IP to graph classification tasks. We then perform experiments on a binary graph classification task from chemistry, mutagen classification, using a user-defined set of induced subgraph enumeration-based queries that are relevant to the task. A mutagenic molecule is a substance, either naturally occurring or synthetic, that increases the frequency of DNA mutations in an organism beyond the normal background rate. These mutations can potentially lead to diseases such as cancer. We believe that such a framework would provide semantic explanations that are insightful to a chemist studying the effect of functional groups (specific subgraphs of a molecular graph) on a molecule's chemical properties, acting as a transparent and trustworthy companion during the scientific discovery process.

Paper Contributions. (1) We propose a framework for creating a user-defined query set for V-IP in the context of graph classification tasks, as well as insight into partially automating the query set creation process. (2) Using this framework, we craft a user-defined query set for a mutagen classification task and apply V-IP to a dataset of molecules labelled as mutagenic or non-mutagenic (Kazius et al., 2005) in order to obtain interpretable predictions. (3) We then empirically compare our experiment results to black-box GNNs, another interpretable-by-design neural network-based method called GNAN (Bechler-Speicher et al., 2024a), and a decision tree-based method called TREE-G (Bechler-Speicher et al., 2024b). (4) Finally, we qualitatively interpret V-IP's explanations, and provide a valuable direction for future work in interpretable mutagen classification, and graph classification in general.

2 Related Work

Post-Hoc Interpretability for Graph Classification. With black-box neural networks becoming ubiquitous in many ML tasks, methods for trying to explain black-box networks, termed *post-hoc* interpretability methods, have also emerged. These methods are model agnostic, as explanations are produced *after* training time, allowing the ML practitioner to prioritize model accuracy. Although, recent research has criticized these methods for being unreliable, especially in high-stake scenarios (Adebayo et al., 2018; Slack et al., 2020; Rudin, 2019). GNNExplainer (Ying et al., 2019) is a post-hoc method for interpreting the predictions of black-box Graph Neural Networks (GNNs), which have become a powerful tool for graph classification tasks, by identifying a subgraph and a set of node features in the input graph that play a significant role in the model's prediction (Ying et al., 2019).

Interpretable-By-Design Methods for Graph Classification. An alternative interpretability paradigm that has emerged is the *interpretable-by-design* framework, where *user-interpretable* explanations provided by model are inherently aligned with the model's internal reasoning (Chattopadhyay et al., 2023b;a). These user-interpretable explanations should also be semantically relevant to the domain, user, and task, through the usage of words, symbols, patterns etc. The benefit of interpretable-by-design methods is full transparency in the model's decision-making process, although these methods are not model agnostic. Variational Information Pursuit (V-IP) is a neural network-based, interpretable-by-design method that sequentially asks interpretable queries about the data until a prediction can be made with sufficient confidence (Chattopadhyay et al., 2023a). V-IP has previously been applied to image classification, text classification, and medical diagnosis tasks,

but applications to graph classification tasks have not been explored yet, which our work aims to do. Other neural network-based interpretable-by-design methods for graph classification tasks have also recently emerged, such as Graph Neural Additive Networks (GNAN), which is an extension of Generalized Additive Models to graph data (Bechler-Speicher et al., 2024a). Finally, specialized decision tree-based methods have also emerged as interpretable models for graph classification, such as TREE-G (Bechler-Speicher et al., 2024b).

3 BACKGROUND

3.1 USING INFORMATION PURSUIT TO LEARN PREDICTORS BY COMPOSING INTERPRETABLE QUERIES

Let $X : \Omega \to \mathcal{X}$ and $Y : \Omega \to \mathcal{Y}$ be random variables that represent the input data and the corresponding labels, respectively. We can define a *query set* Q of user-defined functions of the data $q : \mathcal{X} \to \mathcal{A}$, called *queries*, where $\mathcal{A}$ is the set of possible answers to a query. Note that $P(Y \mid X)$ is the true conditional label distribution. We assume that $Q(X) := \{q(X) : \forall q \in Q\}$ is a sufficient statistic for Y. More formally, this means that $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$,

$$p(y \mid x) = p(y \mid \{x' \in \mathcal{X} : q(x') = q(x),\, \forall q \in Q\}) \tag{1}$$

Given a query set Q, we can use it to sequentially ask queries about our input data until we can make a prediction about its label with some level of sufficient confidence. Information Pursuit (IP) is a greedy algorithm that chooses queries one at a time in order of mutual information gain (Geman & Jedynak, 1996). We can use IP to compose a sequence of most-informative queries to make predictions about the data (Chattopadhyay et al., 2023b;a). In addition to making accurate predictions about the data using the queries produced by IP, we would also prefer sequences of queries that are shorter.

For some observation of the data $x^{obs} \in \mathcal{X}$ and a query $q \in Q$, let $I(q(X); Y)$ denote the mutual information between answers and labels, and let $I(q(X); Y \mid q_{1:k}(x^{\text{obs}}))$ denote the mutual information between answers and labels conditioned on the history of answers $q_{1:k}(x^{\text{obs}}) := q_1(x^{\text{obs}}), ..., q_k(x^{\text{obs}})$ previously observed by IP using queries $q_1, ..., q_k \in Q$. We can compose a sequence of queries using IP as follows:

$$\begin{aligned} q_1 &:= \text{IP}(\emptyset) = \arg\max_{q \in Q} I(q(X); Y) \\ q_{k+1} &:= \text{IP}(\{(q_i, q_i(x^{obs}))\}_{1:k}) = \arg\max_{q \in Q} I(q(X); Y \mid q_{1:k}(x^{\text{obs}})) \end{aligned} \tag{2}$$

We can terminate the IP algorithm using one of two methods proposed in (Chattopadhyay et al., 2023a),

1. Fixed Budget (FB): Terminate after some fixed L number of iterations. So the final query is given by $q_L := \text{IP}(\{q_i, q_i(x^{obs})\}_{1:L-1}) = \arg\max_{q \in Q} I(q(X); Y \mid q_{1:L-1}(x^{\text{obs}}))$
2. Variable Query Lengths (VQL): Terminate when we reach some sufficient level of confidence. More formally, when $\max_Y P(Y \mid q_{1:k}(x^{obs})) \geq 1 - \epsilon$, where ϵ is a tuneable hyperparameter.

3.2 G-IP: INFORMATION PURSUIT USING GENERATIVE MODELS

Computing the mutual information terms in IP can be challenging in practice. To compute $I(q(X); Y)$ we need the joint distribution $P(q(X), Y)$, and to compute $I(q(X); Y \mid q_{1:k}(x^{\text{obs}}))$ we need the joint distribution $P(q(X), Y \mid q_{1:k}(x^{\text{obs}}))$, which are not straightforward to directly compute for high-dimensional data like images or graphs. Furthermore, these mutual information terms need to be computed $\forall q \in Q$, making it even more challenging. Instead, the joint distributions

needed to compute these terms can be approximated by learning probabilistic generative models, such as Variational Autoencoders (VAEs) (Kingma & Welling, 2014), and use MCMC sampling to estimate the mutual information terms (Chattopadhyay et al., 2023b). Although, this process can be computationally expensive for large query sets and high-dimensional data (Chattopadhyay et al., 2023a).

3.3 V-IP: Variational Characterization of Information Pursuit

Variational Information Pursuit (V-IP) (Chattopadhyay et al., 2023a) is a variational characterization of IP that can be less computationally expensive than using generative models, especially for large query sets and high-dimensional data. Also, it produces more accurate predictions alongside shorter sequences of queries for some tasks (Chattopadhyay et al., 2023a).

Let $\mathcal{H}(x)$ be the set of all possible finite-length sequences of query-answer pairs of the form $\Big(\big(q_1, q_1(x)\big), ..., \big(q_m, q_m(x)\big)\Big)$ or $\emptyset$, for some data point $x \in \mathcal{X}$, where $1 \leq m \leq |Q|$ for a non-empty sequence. Now, let $\mathcal{H} := \bigcup_{x \in \mathcal{X}} \mathcal{H}(x)$ be the set of all possible finite-length sequences of query-answer pairs for any arbitrary data point. A *history* is some $s \in \mathcal{H}$, and the random variable $S : \Omega \to \mathcal{H}$ represents these histories.

Let $g : \mathcal{H} \to Q$ be the *querier* function that outputs the next-most informative query given an arbitrary history as input, and let $f : \mathcal{H} \to \mathcal{P}_{\mathcal{Y}}$ be the *classifier* function that takes an arbitrary history as input and outputs a probability distribution over the labels $\mathcal{Y}$. The objective of V-IP (Chattopadhyay et al., 2023a) can be defined as the optimization problem,

$$\begin{gathered} \min_{f,g} \ \mathbb{E}_{X,S}\Big[D_{\text{KL}}\Big(P(Y \mid X) \parallel \hat{P}(Y \mid q(X), S)\Big)\Big] \\ q := g(S) \in Q \\ \hat{P}(Y \mid q(X), S) := f\Big(\big((q, q(X))\big) \cup S\Big) \end{gathered} \tag{3}$$

Note that $\big((q, q(X))\big) \cup S$ is the *updated history*. It is some random history concatenated to the query-answer pair for some data point, using the next-most informative query produced by the querier function. This updated history is then passed as input to the classifier function to output the label probabilities given this updated history. Intuitively, the goal of the V-IP objective is to find the querier and classifier functions that minimize the KL-Divergence between this posterior label distribution and the true conditional label distribution, in expectation over all data points and histories produced by X and S respectively.

Let f^*, g^* be the optimal classifier and querier functions that minimize the V-IP objective. If the sampling distribution of histories S, denoted P_S, has positive, non-zero probability mass under the histories observed during exact IP, then the optimal querier realizes the exact IP strategy (Chattopadhyay et al., 2023a),

$$\begin{aligned} q_1 &:= g^*(\emptyset) = \arg\max_{q \in Q} I(q(X); Y) \\ q_{k+1} &:= g^*(\{(q_i, q_i(x^{obs}))\}_{1:k}) = \arg\max_{q \in Q} I(q(X); Y \mid q_{1:k}(x^{\text{obs}})) \end{aligned} \tag{4}$$

The V-IP objective can be approximated by parameterizing f, g using the weights θ, η of two neural networks respectively, resulting in a new objective termed *Deep V-IP Objective* (Chattopadhyay et al., 2023a),

$$\begin{gathered} \min_{\theta,\eta} \ \mathbb{E}_{X,S}\Big[D_{\text{KL}}\Big(P(Y \mid X) \parallel \hat{P}_\theta(Y \mid q_\eta(X), S)\Big)\Big] \\ q_\eta := g_\eta(S) \in Q \\ \hat{P}_\theta(Y \mid q_\eta(X), S) := f_\theta\Big(\big((q_\eta, q_\eta(X))\big) \cup S\Big) \end{gathered} \tag{5}$$

When introducing (4), we noted that P_S must have positive non-zero probability mass under histories observed during IP. In practice, we don't know the exact IP strategy, which is why we are trying to approximate it in the first place. Constructing such a P_S thus leads to the "chicken and egg" problem (Chattopadhyay et al., 2023a), where we can't construct the the required P_S since we don't know the exact IP strategy, and we can't observe the exact IP strategy since we don't have the required P_S. V-IP proposes two clever, practical methods for sampling histories (Chattopadhyay et al., 2023a),

1. Initial Random Sampling: First, sample $k \sim \text{Uniform}\{0, 1, ..., |Q|\}$ and $X \sim P_{\text{data}}$. Then, sample k queries from Q uniformly at random, and apply them to X to obtain the history. This method can be used to represent the initial sampling distribution $\mathcal{P}_S^0$. The idea is to use $\mathcal{P}_S^0$ to initially "warm up" the querier during training, although convergence can be quite slow.
2. Subsequent Adaptive Sampling: First, sample $k \sim \text{Uniform}\{0, 1, ..., |Q|\}$ and $X \sim P_{\text{data}}$. Then, using the solution querier q_{η^j} to V-IP with P_S^j as the sampling distribution, recursively generate a sequence of k queries using equation (4) with q_{η^j} as the querier. Apply these queries to X to obtain the history. This method can be used to represent the sampling distribution P_S^{j+1}. The idea is to subsequently use this method during training steps to fine-tune the querier, also resulting in faster convergence than Initial Random Sampling.

4 Methods

4.1 Interpretable Graph Classification

Many different types of data, such as social networks, transportation networks, molecules etc., can be represented as graphs. Graph Neural Networks (GNNs) have become ubiquitous for graph classification tasks due to their ability to learn nuanced patterns in graph data without the need for feature engineering, and produce high-accuracy predictions. Although, GNNs are black-box and lack interpretability in their predictions. We can apply an interpretable-by-design method, like V-IP, to produce interpretable predictions for graph classification tasks.

Mutagen Classification. In chemistry, a molecule is called *mutagenic* if it can alter the genetic material (such as DNA) of an organism, increasing the frequency of future genetic mutations, which can cause cancer in animals. Else, it is called *non-mutagenic*. A molecule can be represented as an undirected graph with node attributes (atom types)[2] and edge attributes (number of bonds between two atoms). Chemists are interested in studying which *functional groups* (specific subgraphs of a molecular graph) contribute to a molecule being classified as mutagenic or not. For instance, the presence of the *benzene*, NH_2, and NO_2 functional groups in a molecule are known to induce mutagenic effects (See example in Figure 2) (Debnath et al., 1991). Using a labelled dataset, one can perform supervised learning to model this binary graph classification task. Black-box GNNs can be used for mutagen classification, but lack interpretability. Thus, it would be valuable to use V-IP to explain a model's mutagen predictions using subgraph-based queries that provide explanations about functional groups contributing to the mutagenic effects of a molecule.

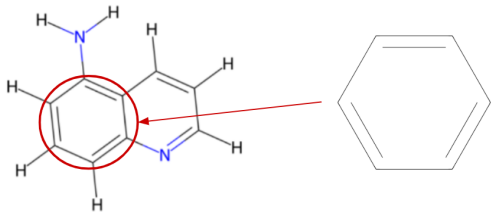


Figure 2: Example molecular graph of a benzene functional group (a ring of six carbon atoms) in a mutagenic molecule.

4.2 Induced Subgraph Enumeration-Based Query Set for Graph Classification using V-IP

As mentioned previously, it would be valuable to produce interpretable predictions by composing a sequence of induced subgraph-based queries using V-IP, especially for the mutagen classification

[2]In a molecular graph, nodes with the Carbon (C) atom type are not labelled. Nodes without an atom type symbol in the graph are implicitly Carbon (C) atoms.

task. We thus create a query set where a query asks about the number of occurrences of a certain induced subgraph in a graph, and the answer is a binary "Yes" or "No". for some graphs G_1, G_2 and some number of occurrences $n \in \mathbb{N}$ we wish to ask about, an induced subgraph enumeration query and its answer have the form,

$$q = \text{"Are there } n \text{ occurences of induced subgraph } G_1\text{?"} \\ q(G_2) \in \{\text{"Yes"}, \text{"No"}\} \tag{6}$$

This formulation may not seem the most natural, as one may naturally define a query "How many occurrences of induced subgraph G_1?" and have the answer be the number of occurrences, $g(G_2) \in \mathbb{N}$. The existing V-IP implementation we were working with from (Chattopadhyay et al., 2023a), already had mechanisms in place for handling binary answers to queries. Due to the limited time and scope of this project, we decided to use the existing binary answer implementation for our subgraph enumeration-based query set. Future improvements on this method should definitely experiment with non-binary answers to these queries, as we hypothesize that V-IP would produce much shorter sequences of queries and ask no redundant queries, which are important criteria for making the sequences more interpretable to the user.

User-Defined Query Set. A user, ideally someone with domain knowledge in the graph classification task being solved, crafts a set of graphs $\mathcal{G}$ whose number of occurrences in a data sample $x \in \mathcal{X}$ will be queried for using query-answer Formulation (6). The set $\mathcal{G}$ should generate a query set that will be used in V-IP to produce interpretable predictions which are relevant to the task being solved. For instance, for mutagen classification, a chemist may define $\mathcal{G}$ to be a set of functional groups whose mutagenic effects they would like to study. Additionally, the query set generated by $\mathcal{G}$ should be sufficient for solving the task, as formalized in Equation (1), which in practice is challenging to do and thus is approximated (Chattopadhyay et al., 2023b). An Induced Subgraph Isomorphism Enumeration Algorithm is then used to count the number of occurrences of some $g \in \mathcal{G}$ for some data point x in the graph dataset $\mathcal{D}$ and generate the query set Q using Algorithm (1). Creating a query set in this way gives full control to the user over what the set of possible prediction explanations will be, which can make explanations more interpretable.

Algorithm 1 Generate Query Set

Require: Set of graphs $\mathcal{G}$, graph dataset $\mathcal{D}$
Ensure: Query set Q

```
function GenerateQuerySet(𝒢, 𝒟)
    Q ← ∅
    for x in 𝒟 do
        for g in 𝒢 do
            n ← InducedSubgraphIsomorphismEnumeration(g, x)
            q ← "Are there n occurrences of g?"
            Q ← Q ∪ {q}
        end for
    end for
    return Q
end function
```

Automatically Generated Query Set In some cases, it may be useful to automatically generate an induced subgraph enumeration-based query set from the data (see Algorithm 2), where doing so creates a set of queries that are still interpretable to the user and relevant to the task. In the context of mutagen classification, it may be valuable to look beyond well-studied functional groups, and instead include all possible induced subgraphs[3] that are present in the dataset which are chemically valid (i.e. the attributes of the nodes and edges present in the subgraph produce a molecule that is in accordance to chemistry laws). The explanations would thus not consist of functional group names, but instead the visualizations of the subgraph, or alternatively using "SMILES arbitrary target specification" (SMARTS), which is a language known to chemists for specifying substructures in molecules.

[3]This includes variations in node and edge attributes as well, not just structurally.

Crafting a query set in this manner possesses some challenges, namely that there could be a very large number of possible induced subgraphs that can exist in a dataset of graphs. First of all, the graph isomorphism problem, which is known to be in the low hierarchy of the NP class, arises when checking for duplicates while getting all induced subgraphs in the dataset. In practice, getting all induced subgraphs is only computed once when initially generating the query set, and graph isomorphism algorithms are relatively efficient nowadays. Although, consideration is needed for the size and density of the graphs in the dataset, the size of the dataset itself, and the computational resources available. Second, generating a query set in this manner can produce a very large query set. When employing V-IP with this query set, neural network architectures that can handle very large inputs, possibly Transformers, must be employed. Although, we can add constraints to the type of induced subgraphs that we allow in our query set in order to shrink its size. For instance, we can restrict the subgraphs to certain numbers of nodes and edges, or to possess certain structural features, shapes, node/edge attributes etc. For mutagen classification, it may be relevant to only look at subgraphs of smaller sizes or with certain structural features. Furthermore, we may restrict to subgraphs that are chemically valid.

A more data-driven way of shrinking the query set of all possible induced subgraphs to some smaller representative set, could be to use clustering algorithms. For instance, one may first convert the subgraphs to a numerical input such as a pairwise similarity matrix using a graph kernel like the Weisfeiler-Lehman (WL) kernel (Shervashidze et al., 2011), or an embedding vector using a graph embedding algorithm. Then one can apply a clustering algorithm of choice, and select subgraphs from each cluster that are most frequently occurring in the dataset (or via some other selection criterion), producing a query set with subgraphs that are "representative" of the dataset.

Throughout the rest of this paper, we only work with a user-defined query set. Applying V-IP to graph classification tasks with an automatically generated query set is left for future work.

Algorithm 2 Automatically generate Graph Set $\mathcal{G}$ subject to constraints C

Require: Graph dataset $\mathcal{D}$, list of constraints C
Ensure: Set of graphs $\mathcal{G}$

```
function GenerateGraphSet(D, C)
    G ← ∅
    for x in D do
        G_x ← GetAllNodeInducedSubgraphs(x, C)        ▷ Subject to constraints
        G ← G ∪ G_x
    end for
    G ← RemoveDuplicateGraphs(G)
    return G
end function
```

5 Experiments

5.1 Dataset & Query Set Pre-processing

We applied the methods discussed previously to a mutagen classification task. Namely, we used the *Mutagenicity* dataset (Kazius et al., 2005) which consists of 4,337 molecules labelled as mutagenic or non-mutagenic. We created a User-Defined Query Set of size $|Q| = 407$ following the method in Section 4.2. The graph set $\mathcal{G}$ used to generate Q consists of all functional groups defined in the `Chem.Fragments` module from the ***RDKit*** Cheminformatics Python library (rdk). A query and its answer are encoded as follows,

$$q = \text{"<functional\_group\_name>=<count>?"} \qquad (7)$$
$$q(\text{molecule}) \in \{1, -1\}$$

where $1, -1$ represent "Yes", "No" answers respectively. For instance, the query "benzene=2?" asks "Are there 2 occurrences of the benzene subgraph?". The query set is stored in a 1D array of size

$|Q|$ with the i^{th} query being stored at the i^{th} position in the array. For a single data point, the answers (1 or -1) to all queries are stored in a 1D array of size $|Q|$, with the i^{th} element in the answer array representing the answer to the i^{th} element in the query set array. Answers to all queries are precomputed for all data points in the dataset, before training, using RDKit's functional group enumerator functions in the `Chem.Fragments` module (which are Induced Subgraph Isomorphism Enumeration algorithms in the back-end, specialized for molecular data) (rdk).

In minimizing the Deep V-IP Objective (5), histories of query-answer pairs are observed. For a single data point, a history is implemented as the precomputed answer array for that data point, with the queries that are *not* included in the history being masked out with 0's, as is done in (Chattopadhyay et al., 2023a). Note that during training, V-IP never sees the graph data structure of a data point, it only sees the masked history array and the corresponding binary label for the data point (mutagenic or non-mutagenic).

5.2 V-IP Training Setup

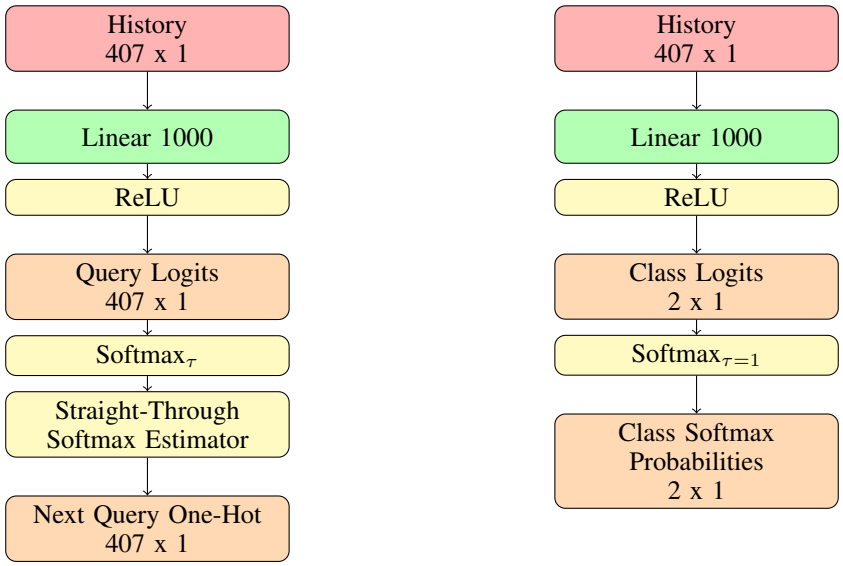
Network Architecture & Train Iteration. For the neural networks used to parameterize the querier and classifier functions in the Deep V-IP Objective (5), we opted for simple MLP architectures with a single hidden linear layer and ReLU activation (Figure 3). In the forward pass, for a single data point, the Querier network takes an arbitrary, masked history as input, implemented as described in Section 5.1, and outputs a one-hot vector q_{onehot}. The next-most informative query given the history is the query at position $i = \texttt{argmax}\ q_{onehot}$ of the query set array. The history is updated by unmasking the i^{th} element of the history and replacing it with the i^{th} element of the data point's answer array. The updated history is passed to the Classifier network, which outputs class logits for the data point, using only the query-answers observed so far in the updated history. These class logits are the criterion used to minimize the loss. In the backward pass, the gradients flow through both the classifier *and* the querier. A *straight-through* softmax gradient estimator (Paulus et al., 2021) is used in the querier during the backward pass, since the hard probability one-hot vectors it outputs during the forward pass are not differentiable. In a single training epoch, this process is conducted for each data point in the training set, sampling the initial masked history for the data point using Random Sampling or Adaptive Sampling (Section 3.3).

Optimization Scheme. The optimization scheme used is nearly identical to that in (Chattopadhyay et al., 2023a). Minimizing the Deep V-IP Objective (5) is equivalent to minimizing the Cross-Entropy Loss (Chattopadhyay et al., 2023a). Thus, in practice we minimize a single Cross-Entropy Loss in mini-batches using the weights of the Querier and Classifier networks, optimized with Adam (Kingma & Ba, 2014), using a learning rate `lr=1e-4`, `betas=(0.9, 0.999)`, `weight_decay=0`, and `amdgrad=True`. Additionally, a Cosine Annealing learning rate scheduler (Loshchilov & Hutter, 2017) was employed, with `T_max=500`. After each epoch, the training set is also randomly shuffled. The final models used in the experiments were first trained for 500 epochs using Random Sampling, and then trained for another 500 epochs using Adaptive Sampling, each using `batch_size=128`. In the straight-through softmax estimator, the softmax temperature τ is linearly annealed from 1.0 to 0.2 over the 500 epochs in each of the sampling stages.

Implementation Details. All the experiments are implemented in Python using Pytorch (Paszke et al., 2019) version 2.4.1, and all training is done on a computer with a 20-core, 4.70GHz, 12th Gen Intel(R) Core(TM) i7-12700H CPU, 1 NVIDIA RTX A1000 Laptop GPU with 4GB of VRAM, and 32GB of RAM.

5.3 Results

Comparison to Baselines. Mean test accuracy on the Mutagenicity dataset in a 10-fold cross validation was obtained for V-IP with two different termination criteria, Fixed Budget (FB) of 20 queries and Variable Query Lengths (VQL) with a confidence threshold of 0.85 (Figure 4). For V-IP with VQL, query sequences had a mean length of 11.9 ± 1.0 and a mean standard deviation of 7.3 ± 0.4. The results were compared to 10-fold cross validation mean test accuracies of black-box GNNs (GraphConv, GraphSAGE, GIN, GATv2, GTransformer, FSGNN) (Bechler-Speicher et al., 2024a), the decision tree-based method TREE-G (Bechler-Speicher et al., 2024b), and an interpretable-by-design neural network-based method called Graph Neural Additive Network (GNAN) (Bechler-Speicher et al., 2024a).



(a) Querier network **(b)** Classifier network

Figure 3: Neural network architectures for (a) the querier and (b) the classifier used in the experiments.

Our V-IP models outperform all the black-box GNN methods by $\sim 2\% - 10\%$, as well as the other interpretable-by-design neural network method, GNAN, by about $\sim 3\% - 4\%$. Both V-IP methods fall short of TREE-G by about $\sim 7\% - 8\%$. The differences in test accuracies between the non-black-box methods (TREE-G, GNAN, V-IP) could be explained by differences in the query sets or data features used to train these models. It could be the case that TREE-G just trained their decision trees with better features than the query set that we used for V-IP. Further investigation into the features used is needed in order to obtain a more rigorous comparison between these models. Furthermore, V-IP also tries to minimize query sequence length, which could themselves be represented as a decision tree. It would be useful to compare the depth of the TREE-G decision trees to the that of the V-IP query sequence decision trees in order to compare the interpretability of these methods.

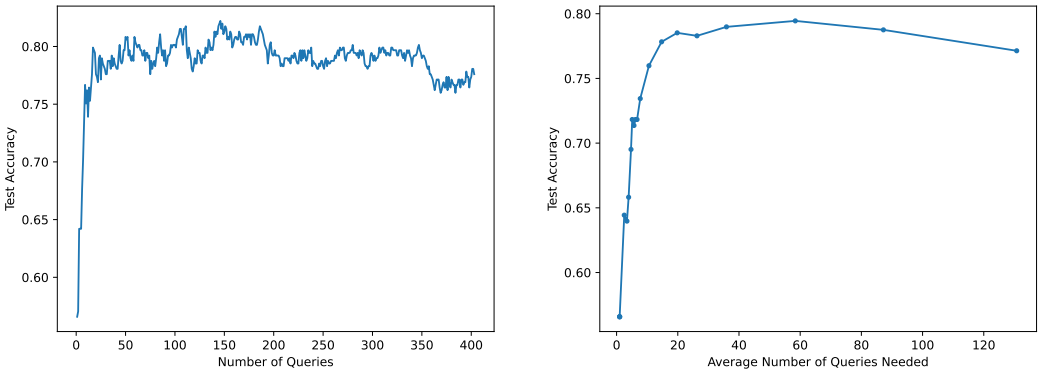
Figure 4: Test accuracy obtained on Mutagenicity dataset using a 10-fold cross validation.

Model	**Test Accuracy**
GraphConv	64.3 ± 1.7
GraphSAGE	64.1 ± 0.3
GIN	69.4 ± 1.2
GATv2	72.0 ± 0.9
GTransformer	73.1 ± 0.9
FSGNN	66.9 ± 1.5
TREE-G	$\mathbf{83.0 \pm 1.7}$
GNAN	72.2 ± 1.0
V-IP (FB=20)	76.8 ± 2.1
V-IP (VQL)	76.4 ± 2.1

Finally, one fold of the 10-fold cross validation, using the training setup described in Section 5.2, took around 45 minutes to train. A rigorous comparison between the training times of interpretable-by-design and black-box neural networks is left for future work.

Comparison between V-IP with FB and VQL. In our experiments, V-IP with VQL performed $\sim 1\%$ worse than with a FB of 20 queries. The performance of VQL could be improved by fine-tuning the confidence threshold hyperparameter. Observe in Figure 5b how by increasing the confidence threshold, the query sequence lengths become longer and the test accuracy increases, until a plateau at around 20 queries. Similarly in Figure 5a for FB, increasing the number of queries increases test accuracy until it reaches a plateau, and increasing test accuracy further requires a significant amount of queries, which sacrifices interpretability. Furthermore, we hypothesize that by training for more than 500 epochs, VQL will produce shorter query sequence lengths with higher test accuracies than FB. Due to the limited time and scope of the project, this was left for future work.

Qualitative Analysis of V-IP Explanations with VQL. The V-IP predictions and query-answer sequences for a data point can be visualized using a heatmap of the class softmax probabilities given



(a) Fixed Budget

(b) Variable Query Lengths

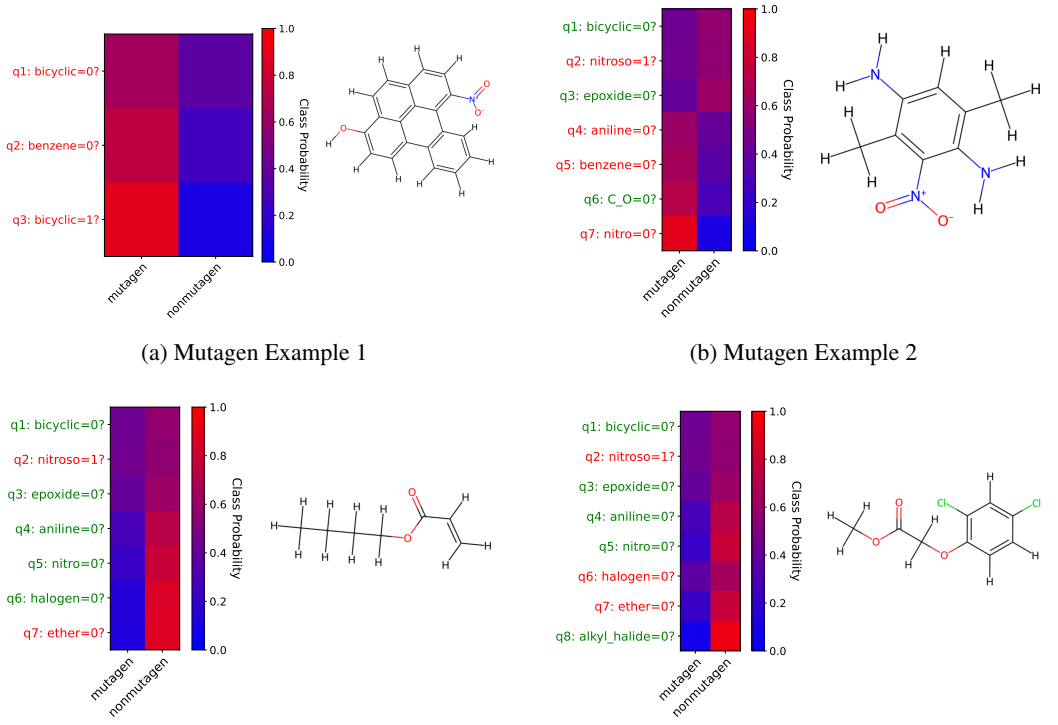
Figure 5: A comparison between V-IP termination criteria, namely (a) Fixed Budget (FB) and (b) Variable Query Lengths (VQL), and their effect on test accuracy as query sequence lengths change. These figures are for one fold of the 10-fold cross validation split.

the history of queries asked until a certain row in the heatmap. The querying terminates when one of the class probabilities surpasses the chosen confidence threshold which in Figure 6 is 0.85. In the mutagen examples (6a, 6b), queries about *benzene*, NO_2, and *bicyclic* (two fused rings, such as two benzene) functional groups are asked in the query sequence. V-IP seems to have learned that these queries are very informative, aligning with the intuition that these functional groups are known to induce mutagenic effects in molecules (Debnath et al., 1991). The prevalence of benzene and bicyclic queries in (6a) also aligns with the explanations that GNNExplainer and GNAN produced for this same molecule (Ying et al., 2019; Bechler-Speicher et al., 2024a). The non-mutagenic examples (6c, 6d) have many queries about the *lack* of presence of certain functional groups, possibly indicating that the most-informative queries involve functional groups commonly found in mutagenic molecules. Intuitively, it is most informative to rule out mutagenic molecules first, rather than non-mutagenic first. This can also be supported by the fact that query sequence lengths for mutagenic molecules tend to be shorter than those of non-mutagenic molecules. Also, V-IP correctly identifies (6d) as non-mutagenic even though it contains one benzene, although it requires more queries to reach sufficient confidence. A much more in-depth analysis of these explanations, incorporating expertise in chemistry, would be very valuable, and is left for future work.

6 Future Work

There are many improvements that can be made to our work in regards to both accuracy and interpretability. First of all, it would be most beneficial to conduct future research alongside an expert in the domain or graph classification task of interest. A lot of research in the Explainable AI field is produced by mathematicians, computer scientists, and data scientists that lack domain knowledge in the datasets and tasks they are working with, and rather are algorithm-focused. We believe that domain-focused research is crucial in the Explainable AI field, as interpretable-by-design methods like V-IP (Chattopadhyay et al., 2023a) are rooted in the idea of producing interpretable predictions that are semantically useful to the user and task, which ultimately requires domain expertise. For mutagen classification, for instance, a chemist could aid in the creation of a more sufficient user-defined query set, as well as interpreting and rating the quality of V-IP explanations in greater depth, fine-tuning the model as needed. One could then expand this research to graph datasets from other domains, such as proteins, transportation networks, transaction networks for fraud detection etc. also incorporating domain expertise.

We believe that future work on automatically generating query sets, as introduced in Section 4.2, could aid in the creation of a sufficient query set, as doing so is not trivial. We believe this may be beneficial for mutagen classification, among other graph classification tasks as well, and should be explored further in this context. Again, working alongside a domain expert would ensure that interpretability is not sacrificed in the process, and the explanations still possess relevant semantic



(a) Mutagen Example 1

(b) Mutagen Example 2

(c) Non-Mutagen Example 1

(d) Non-Mutagen Example 2

Figure 6: Each row in a heatmap shows the posterior probability of the class label given the history of query-answers on the left of the heatmap (Green = "Yes", Red = "No"). The sequence of queries terminates once a class probability is greater than the confidence threshold 0.85. Two examples (a) and (b) are for mutagenic molecules, and the other two (c) and (d) for non-mutagenic molecules.

information. If query sets become too large for an MLP neural network architecture to handle, one could experiment with Transformer architectures. Finally, a more rigorous and systematic comparison between V-IP and non-black-box models (GNAN, TREE-G) for graph classification tasks is warranted, keeping in mind interpretability and the data features used for training, rather than just test accuracy. In addition, exploring the differences in training speed of black-box and interpretable-by-design methods would also be very valuable, especially for larger graph datasets that may require lengthier query set pre-processing times.

7 Conclusion

V-IP is an interpretable-by-design method that was previously applied to image classification, text classification, and medical diagnosis tasks. We introduced a framework for applying V-IP to graph classification tasks, using induced subgraph enumeration-based query sets. Using this framework, we created a reasonably sufficient dataset for solving a mutagen classification task. We then quantitatively compared test accuracy results on this task with other interpretable-by-design methods and black-box GNNs. V-IP surpassed all black-box baselines in accuracy, as well as GNANs, another neural network-based interpretable-by-design method. Finally, through qualitative analysis of V-IP's explanations for mutagen classification, we highlighted the interpretability of the method in practice, and provided a direction for future domain-centric research in Explainable AI.

References

RDKit: Open-source cheminformatics. URL https://doi.org/10.5281/zenodo.14535873. https://www.rdkit.org/.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf.

Maya Bechler-Speicher, Amir Globerson, and Ran Gilad-Bachrach. The intelligible and effective graph neural additive network. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=SKY1ScUTwA.

Maya Bechler-Speicher, Amir Globerson, and Ran Gilad-Bachrach. Tree-g: Decision trees contesting graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38 (10):11032–11042, Mar. 2024b. doi: 10.1609/aaai.v38i10.28979. URL https://ojs.aaai.org/index.php/AAAI/article/view/28979.

Aditya Chattopadhyay, Kwan Ho Ryan Chan, Benjamin David Haeffele, Donald Geman, and Rene Vidal. Variational information pursuit for interpretable predictions. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=77lSWa-Tm3Z.

Aditya Chattopadhyay, Stewart Slocum, Benjamin D. Haeffele, René Vidal, and Donald Geman. Interpretable by design: Learning predictors by composing interpretable queries. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(6):7430–7443, June 2023b. ISSN 0162-8828. doi: 10.1109/TPAMI.2022.3225162. URL https://doi.org/10.1109/TPAMI.2022.3225162.

Asim Kumar Debnath, Rosa L. Lopez de Compadre, Gargi Debnath, Alan J. Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry*, 34(2):786–797, 1991. doi: 10.1021/jm00106a046. URL https://doi.org/10.1021/jm00106a046.

Donald Geman and Bruno Jedynak. An active testing model for tracking roads in satellite images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(1):1–14, January 1996. ISSN 0162-8828. doi: 10.1109/34.476006. URL https://doi.org/10.1109/34.476006.

Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of Medicinal Chemistry*, 48(1):312–320, 2005. doi: 10.1021/jm040835a. URL https://doi.org/10.1021/jm040835a. PMID: 15634026.

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*, pp. 267–280. Springer-Verlag, Berlin, Heidelberg, 2022. ISBN 978-3-030-28953-9. URL https://doi.org/10.1007/978-3-030-28954-6_14.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Adam 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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Max B Paulus, Chris J. Maddison, and Andreas Krause. Rao-blackwellizing the straight-through gumbel-softmax gradient estimator. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Mk6PZtgAgfq.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 05 2019. doi: 10.1038/s42256-019-0048-x.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.*, 12(null):2539–2561, November 2011. ISSN 1532-4435.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, pp. 180–186, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371100. doi: 10.1145/3375627.3375830. URL https://doi.org/10.1145/3375627.3375830.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/d80b7040b773199015de6d3b4293c8ff-Paper.pdf.