# Improving Concept Extraction in Graph Classification using Hierarchical Pooling

**Jingyi Zhao***
University of Cambridge
jz610@cam.ac.uk

**Yuxuan Ou***
University of Cambridge
yo279@cam.ac.uk

## Abstract

Recent research like GCExplainer has investigated the methods for explaining Graph Neural Networks (GNNs) by providing global concept-based explanations [1]. GCExplainer extracts concepts on node embeddings in the latent space of the last neighbourhood aggregation layer in a GNN. However, for a graph classification task where the label is associated with the entire graph, extracting concepts before the global pooling layer decreases the quality of explanations. To improve the concept extraction for graph classification models, we take advantage of DiffPool [2], a hierarchical graph pooling technique which cluster nodes together and form coarsened graphs in between GNN layers, and propose hierarchical concept extraction. We investigate the interpretability of hierarchical graph pooling by exploring the clustering of the latent space before and after each pooling layer. Furthermore, we propose a feasible method to construct DiffPool models that are explainable by design. We also present a position on what an interpretable graph pooling technique needs.

## Statement of contribution

We, Jingyi Zhao and Yuxuan Ou of the University of Cambridge, jointly declare that our work towards this project has been executed as follows:

- **Jingyi Zhao** contributed by implementing the initial version of DiffPool model, activation space visualization, and CEM guidance strategy. She also contributed by running experiments and writing report.
- **Yuxuan Ou** contributed by implementing DiffPool model, graph visualization and concept visualization, concept interpretating, training and comparing different DiffPool models. She also contributed by running experiments and writing report.
- **Jingyi Zhao** and **Yuxuan Ou** developed all ideas in this work together.

We both independently submit identical copies of this paper, certifying this statement to be correct.

## GitHub repository with commit log

The companion source code for our project may be found at: https://github.com/Stella-zjy/L65-GDL-project.

## 1 Introduction

Graph Neural Networks (GNNs) have emerged as state-of-the-art methods for reasoning about graphs, due to their ability to recursively incorporate both the feature information and structural information

*Equal contribution.

Jingyi Zhao & Yuxuan Ou, Improving Concept Extraction in Graph Classification using Hierarchical Pooling. *Geometric Deep Learning (L65), University of Cambridge*.

from neighbouring nodes [2–5]. Despite the strength, GNNs lack transparency in the sense that a prediction cannot be directly interpreted by humans. Early studies tackle this problem by providing explanations for a trained GNN and its given prediction(s) [5–7]. However, these approaches are local in the sense that they are only able to provide explanations for a single instance or multiple instances instead of globally explaining the model itself.

Recent work like GCExplainer therefore attempts to provide *concept-based* explanations [1]. Concepts refer to small units of information that are human understandable. GCExplainer extracts concepts on node embeddings in the latent space of the last neighbourhood aggregation layer in a GNN. However, for a graph classification task where the label is associated with the entire graph, extracting concepts before the global pooling layer decreases the quality of explanations. Towards understanding how concepts are formed in earlier layers, Hierarchical Explainable Latent Pooling (HELP) provides a hierarchical pooling procedure that is *interpretable-by-design* [8]. Specifically, HELP repeatedly performs clustering on latent space of node embeddings to extract concepts and then pools nodes belonging to the same concept together.

With the same aim of extracting hierarchical concepts in a graph classification model, we leverage the hierarchical pooling strategy of DiffPool [2]. We extract hierarchical concepts according to nodes that are pooled together in each DiffPool layer. We show how higher-level concepts (i.e., concepts extracted in later DiffPool layers) are combined by lower-level concepts (i.e., concepts extracted in early DiffPool layers). By analyzing the commonalities of nodes pooled together, we show that our hierarchical concepts are human-understandable, justifying the interpretability of DiffPool. Nonetheless, we observe that the interpretability of DiffPool is sensitive to model constructions. Taking advantage of the concept encoding and clustering procedure in Concept Encoder Module (CEM) [9], we provide a potential approach to guide the construction of DiffPool models.

Our main contributions are:

- **Hierarchical Concept Extraction Strategy** We propose a DiffPool-based hierarchical concept extraction strategy for studying the interpretability of graph classification. Meanwhile, we present DiffPool in an interpretable manner.
- **CEM-guided DiffPool Model Construction** We propose a CEM-guided strategy for designing the DiffPool model structure. Specifically, we provide a potential approach to decide the number of DiffPool clusters based on the clustering procedure used in CEM.

## 2 Background

### 2.1 DiffPool

DiffPool provides a differentiable pooling module that is able to hierarchically coarsen the graphs in-between GNN layers [2]. The key idea of DiffPool is to learn a cluster assignment matrix indicating how each node will be pooled into a cluster in the next coarsened layer. DiffPool learns two GNNs for node embedding and pooling respectively. The embedding GNN generates new embeddings for input nodes at this layer, and the pooling GNN generates a probabilistic assignment of the input nodes to the new clusters in the coarsened graph at the next layer. Let's denote the two GNNs as $GNN_{embed}$ and $GNN_{pool}$.

Let's denote the node feature matrix and the adjacency matrix at layer $l$ to be $X^{(l)}$ and $A^{(l)}$, the assignment matrix and embedding matrix learnt from the GNNs at layer $l$ to be $S^{(l)}$ and $Z^{(l)}$. Namely, we have

$$Z^{(l)} = GNN_{embed,l}(X^{(l)}, A^{(l)})$$
$$S^{(l)} = softmax(GNN_{pool,l}(X^{(l)}, A^{(l)}))$$

DiffPool then generates the coarsened graph according to the two following equations:

$$X^{(l+1)} = S^{(l)T} Z^{(l)}$$
$$A^{(l+1)} = S^{(l)T} A^{(l)} S^{(l)}$$

### 2.2 Concept-based Explainers for GNNs

The pioneering work in GNN explainability is GNNExplainer [5], which introduces a method to learn a mask for node neighbours and features, highlighting the subgragh and subset of features

that are pivotal in influencing decision-making. However, GNNExplainer essentially offers local explanations, focusing on individual instances and not addressing global interpretability challenges. In response, GCExplainer[1] emerges as the first method promoting concept-based explanations for GNNs, providing the automatic extraction of global concepts. Concepts means small units of information that are human-understandable. While it's hard to give a concrete definition of concept, Automatic Concept-based Explanation (ACE) method [10], which provides concept-based explanations for CNN classifiers, introduces three desired properties for concept-based explanations:

- ***Meaningfulness*** An instance of a concept should be semantically meaningful on its own. For example, a methyl group (CH3) is meaningful in the sense that a methyl group itself contains important information regarding chemical properties. Additionally, different instances within a concept are expected to have similar meanings.
- ***Coherence*** Instances of a concept should be perceptually similar to each other while being different from instances of other concepts.
- ***Importance*** A concept is expected to be important for the prediction of a class in the sense that its presence is necessary for the true prediction of samples in that class.

Adopting the same idea of concept-based explanations, GCExplainer suggests a strategy for concept discovery in GNNs by defining a mapping from the activation space to the concept space through clustering. Specifically, GCExplainer employs $k$-Means clustering on the activation space, with each cluster representing a distinct concept. Notably, both GNNExplainer and GCExplainer only provide post-hoc explanations, instead of making the model itself more interpretable.

### 2.3 Concept Encoder Module (CEM)

Concept Encoder Module (CEM) provides a differentiable approach which makes GNNs explainable by design [9]. CEM extracts a set of concepts and then uses these to solve a classification task. A concept encoding method is proposed to extract node-level clusters corresponding to concepts. Specifically, the clustering (which we may refer to as CEM clustering in the later part of this report) is performed by a normalized softmax activation on the node-level embeddings.

Let's denote $x_i$ to be the node embeddings for node $i$. Suppose $x_i$ has $m$ entries $x_{ij}$'s for $j \in \{1, 2, \ldots, m\}$, CEM first calculates a fuzzy encoding $q_i \in [0, 1]^m$ for node $i$ via the normalized softmax activation:

$$\hat{q_i} = \frac{\exp x_i}{\sum_{j=1}^{m} \exp x_{ij}}, q_i = \frac{\hat{q_i}}{\max_i \hat{q_i} + \epsilon}, \epsilon > 0$$

CEM then clusters nodes together depending on the Booleanized fuzzy encoding $r_i \in \{0, 1\}^m$:

$$r_{ij} = \begin{cases} 1 & \text{if } q_{ij} > \tau \\ 0 & \text{otherwise} \end{cases}$$

where $\tau \in [0, 1]$ is conventionally set to be 0.5. In particular, two nodes belong to the same cluster if and only if their Booleanized encodings $r_a$ and $r_b$ are exactly the same. The number of clusters in CEM clustering is the number of unique values of $r_i$'s.

## 3 Concept Extraction via DiffPool

### 3.1 Overview of Hierarchical Concept Extraction

The key idea of our hierarchical concept extraction builds upon the hierarchical pooling nature of DiffPool. The upper half of Figure 1 shows the decomposed model structure of our graph classification model. We use three DiffPool layers to coarsen the input graphs three times across the model. It's natural to investigate the latent space before and after each DiffPool layer and figure out which nodes are pooled together. In our three-DiffPool-layer model, we thus extract four levels of concepts which are node concepts from the latent space before the first DiffPool layer, first DiffPool concepts from the latent space after the first DiffPool layer, second DiffPool concepts from the latent space after the second DiffPool layer, and third DiffPool concepts from the latent space after the third DiffPool layer. We ignore the latent space before the second (or third) DiffPool layer because they are just the resulting latent space of the latent space after the first (or second) DiffPool layer when undergoing several GNN layers and contain no information of graph pooling.

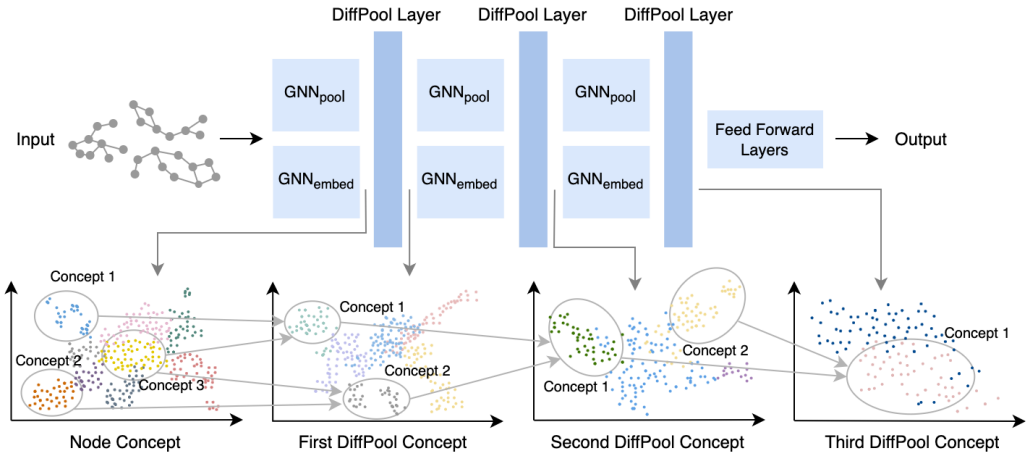



**Figure 1: Overview of Hierarchical Concept Extraction.** The upper half of the figure shows the model structure we used for our graph classification task. We extract node concepts from the latent space before the first DiffPool layer, and extract higher-level concepts after the first DiffPool layer, after the second DiffPool layer, and after the third DiffPool layer. An illustration of concept hierarchy is shown on the hypothetical latent space. Namely, higher-level concepts (concepts extracted from later layers) are dominated by different lower-level concepts (concepts extracted from early layers). Notably, one higher-level concept is usually dominated by several lower-level concepts, and one lower-level concept can contribute to more than one higher-level concept.

We extract node concepts following the unsupervised clustering method as is proposed in GCExplainer [1]. Namely, we perform $k$-Means clustering on the raw activation space of node embeddings right before the first DiffPool layer, where each of the $k$ clusters formed represents a node concept. In this way, nodes belonging to the same cluster should have similar properties across all input graphs, which meets the expectation of a global explanation. For higher-level concepts, we leverage the hierarchical pooling of DiffPool and extract concepts via DiffPool clusters. For a new node in the new coarsened graph formed by a DiffPool layer, it's pooled from several nodes having similar feature and structural information in its original graph. Thus those new nodes belonging to the same DiffPool cluster should also share common characteristics, which we will discuss in detail in Section 3.2. Similarly, instances (i.e., new nodes formed by DiffPool layers) belonging to the same DiffPool cluster come from all input graphs, hence enabling a global explanation.

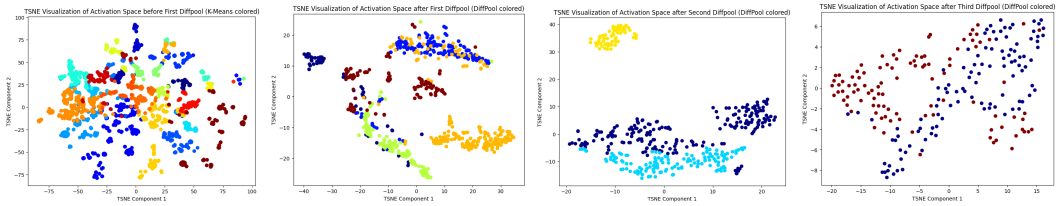
In our model construction, each GNN block consists of 3 Graph Convolutional Network (GCN) Layers (both the GNN pooling block and the GNN embedding block). Each DiffPool layer generates coarsened graphs by pooling nodes in a graph to a selected number of new nodes. The final feed-forward block contains two linear layers with ReLU activation. Each neural network layer has 32 hidden units. We set the number of new nodes to be 8, 4, and 2 for the first, second, and third DiffPool layers respectively. Our model selection will be further discussed and justified in Section 4. All experiment results and visualizations presented in the rest of the report are produced under the above setting (if without further specification).

All the experiments in this report are conducted on the Mutagenicity dataset [11]. The prediction task is a binary graph classification on whether the molecule is mutagenic.

### 3.2 Hierarchical Concepts Extracted by DiffPool

Figure 2 shows the aforementioned four activation space with data points colored via corresponding concept membership. In order to visualize the activation space, we apply TSNE dimension reduction to the raw activation space, but the actual clustering is performed on the raw activation space (i.e., without dimension reduction). The dimension reduction technique is only used for illustration purpose, while this also indicates that the figures we see suffer from loss of information, but such loss should be tolerable. Figure 2a shows the node concept extracted using $k$-Means clustering with $k = 30$ (further investigation of different choices of $k$ is discussed in Appendix B). Note that for the

first (or second) DiffPool concept, although we set the DiffPool cluster number to be 8 (or 4), the actual cluster number turns out to be 5 (or 3) (i.e., only 5 (or 3) colors appear on the activation space).




**(a)** node concept **(b)** first DiffPool concept **(c)** second DiffPool concept **(d)** third DiffPool concept

**Figure 2: Concept Extraction from Activation Space.** Images from left to right are concept extraction from the activation space before the first DiffPool layer, after the first DiffPool layer, after the second DiffPool layer, and after the third DiffPool layer. Data points are colored by concept membership.

Data points in Figure 2a are clustered nicely due to the clustering nature of $k$-Means. Judging from the coloring pattern of Figure 2c and 2d, we observe that the second and third DiffPool concepts are also nicely clustered. In Figure 2b, the coloring pattern is not that separated and data points belonging to different concepts sometimes appear to be overlapping with each other. However, the majority of navy-blue, blue, yellow, and green points are clearly clustered, and the locations of most brown points can also be spotted in the middle of the space. We hence consider the first DiffPool concept to be satisfyingly clustered as well. Such clustering pattern implies that DiffPool is able to cluster similar nodes together, and that our choosing DiffPool clusters to be concepts is reasonable.

To assess the concepts identified by DiffPool, we visualize the concepts extracted and evaluate them according to the standards of concepts proposed in ACE[10]. We select the data points nearest to the centroid of each DiffPool cluster in the raw activation space, focusing on the first and second DiffPool layers. The visualization for the first DiffPool cluster, depicted in Figure 3, demonstrates that DiffPool is capable of extracting concepts that are interpretable from a human perspective. The identified concepts are as follows. The first concept is a carbon-oxygen structure, including a carbon (C) and oxygen (O) atom, with the carbon atom participating in a hexagonal ring structure. The carbon atom is assigned with node concept 29 and the oxygen atom is assigned with 24. This concept highlights the common aromatic structure found in many organic molecules. The second concept represents hydrogen (H) atoms in the substructures of methylene (CH2) and methyl (CH3) groups, indicating the identification of common hydrocarbon components. The hydrogen atom is assigned with node concept 3, 26 and 1. The third concept showcases a bromine (Br) atom attached to a carbon atom. The bromine atom is assigned with node concept 27. The fourth concept involves a carbon atom bonded to an oxygen atom, without the inclusion in a hexagonal ring, pointing towards functional groups like carbonyls or alcohols. The oxygen atom is also assigned with node concept 24. The fifth concept contains a nitrogen (N) atom connected to two hydrogen(H) atoms and an OH group. All the three atoms in the NH2 group is assigned with 6 and the atoms in OH group is assigned with 11. In all, we think the concepts extracted are meaningful in chemistry context and they are coherent within the concept and distinct with other concepts.

The concepts extracted by the second layer of DiffPool are higher-level concepts, with each deeper concept comprising lower-level concepts from the previous layer. This layer continues to demonstrate a observable pattern in the extracted concepts, emphasizing the model's ability to identify complex and layered structures. The first concept is derived from concept 7 of the first DiffPool layer, which is adjacent to another concept. The second concept originates from concept 5 of the first DiffPool layer and is adjacent to two other concepts. The third concept also evolves from concept 5, similar to the second hierarchical concept, but it expands further by connecting to three additional concepts. This progression underscores the model's capacity to extract more complex and nuanced hierarchical structures. Notably, here concept 7 refers to the concept in the last row in Figure 3 and concept 5 refers to the concept in the fourth row. Through the visualization of these concepts, it is evident that DiffPool is adept at extracting hierarchical concepts.

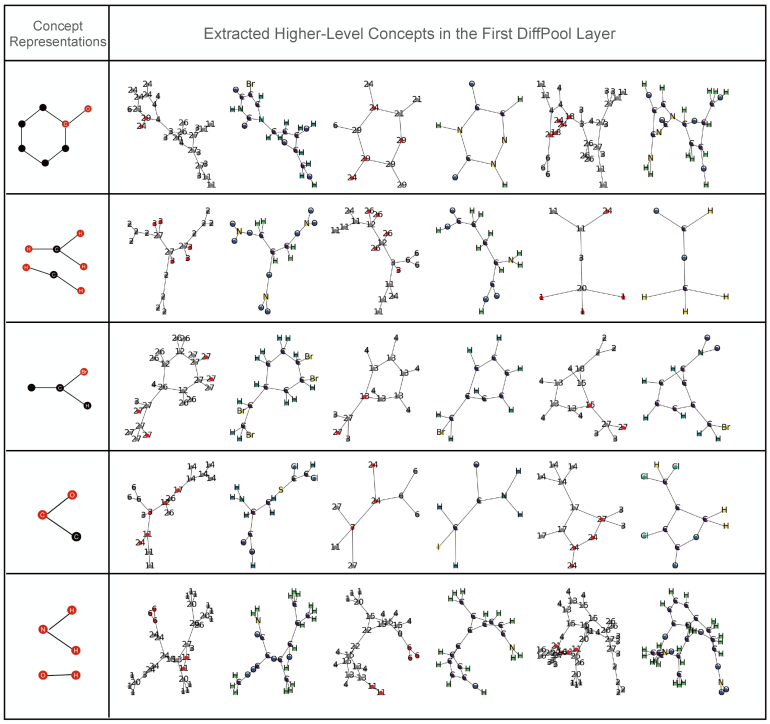



**Figure 3: First DiffPool Concept Visualization.** Each row represents a distinct higher-level concept. Although eight clusters were designated in the first DiffPool layer, only five were utilized. The first column displays the concept representation of the extracted concepts. In the second column, each graph is presented twice, with different coloring and labeling. In the left graph, the numbers on each node indicate the node concepts assigned by K-Means following DiffPool. The red nodes highlight the nodes pooled together to form the higher-level concept. In the right graph, the characters on each node denote the atom type.

### 3.3 Concept Interpretability Comparison

In the preceding section, we demonstrated that DiffPool could facilitate the extraction of higher-level concepts. To show that it is beneficial for hierarchical concept extraction, in this section, we show that DiffPool can enhance the extraction of node concepts. To compare the quality of node concepts, we employ the concept completeness score as is defined in [12]. The score is derived from the prediction of a decision tree which uses the one-hot encoding of concepts assigned to an input instance $x$ to predict the output label $y$. The formula for the concept completeness score is:

$$\eta_f(c_1, \ldots, c_m) = \frac{g(c(x)) - a_r}{f(x) - a_r}$$

Here, $f(x)$ represents the graph classification model with $x$ being the graph input, $c(x)$ denotes the one-hot encoding of concepts, and $g$ is a decision tree classifier. The term $a_r$ corresponds to the accuracy of a random guess. We compare the concept completeness score of DiffPool with that of CEM. Assigning 30 node concepts to both networks, we present their concept completeness scores in Table 1.

The concept completeness scores indicate that DiffPool can help node concept assignment when number of node concepts are the same. In turn, the improvement in node concept extraction also benefits the formation of higher-level concepts.

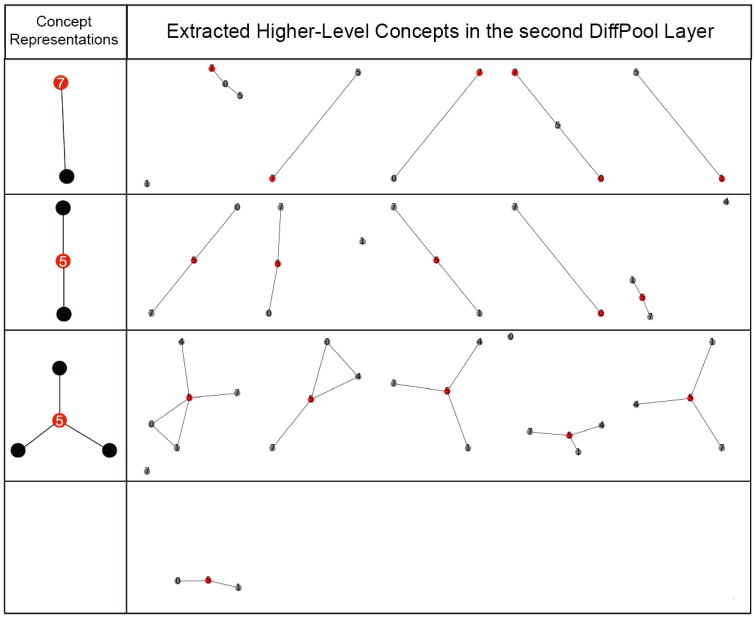



**Figure 4: Second DiffPool Concept Visualization.** There are 3 clusters in the second DiffPool layer, where the cluster that has only one graph is ignored. The first column is the concept representation of the extracted concept. The graphs are the new graphs after first DiffPPool and the numbers on each nose represent concepts assigned by the first DiffPool layer.The red nodes represent the nodes that got pulled together in the second DiffPool.

**Table 1:** Concept Completeness Scores for Node Concepts of CEM and DiffPool

| Method | Concept Completeness Score |
|---|---|
| CEM | 0.477 |
| DiffPool | **0.626** |

# 4 CEM Guidance for DiffPool Model Construction

## 4.1 Influence of Model Construction on DiffPool Interpretability

We observe that the architecture of the DiffPool model is critical for showcasing its capacity to extract hierarchical concepts and enhance interpretability. Not all configurations of the DiffPool model will yield high interpretability. Consider a model comprising only two DiffPool layers: the first with 8 clusters and the second with 4. The node concept completeness scores for this model are presented in Table 2, indicating that a three-layer model is better at extracting node concepts.

**Table 2:** Concept Completeness Scores for Node Concepts of DiffPool Models with Different Number of DiffPool Layers

| DiffPool Layer Number | Concept Completeness Score |
|---|---|
| 2 | 0.583 |
| 3 | **0.626** |

The visualization of hierarchical concept extraction in the second DiffPool layer is depicted in Figure 5. Despite assigning four clusters to this layer, only three are used. Within these clusters, there is an apparent lack of uniform structure and combination of lower-level concepts. Therefore, in terms of the criteria of coherence in ACE, the concepts extracted are not coherent. Hence, determining the

optimal structure of the DiffPool model is of great significance to ensure effective interpretability and the extraction of hierarchical concepts.

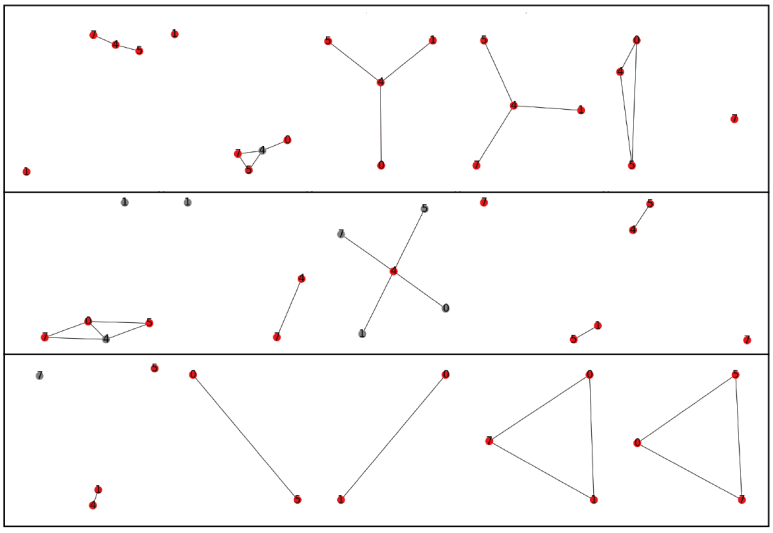


**Figure 5: Second DiffPool Concept Visualization for the Model with 2 DiffPool Layers.** The hierarchical concept representations are not consistent within each cluster.

### 4.2 CEM Guidance on Model Construction

We propose a CEM-guided model construction strategy for selecting the cluster numbers of DiffPool layers. As is discussed in Section 2.3, CEM generates a Booleanized string for the node embeddings of each graph, which serves as a natural clustering criterion. We propose to apply the same CEM clustering on the latent space right before the DiffPool layer and decide the number of clusters for the DiffPool layer based on the cluster number of CEM clustering. To do so, we construct a Vanilla GNN model which has the same GNN structure with DiffPool GNN before the DiffPool layer to be designed. The Vanilla model then replaces the DiffPool layer with a standard global pooling layer (e.g., global mean pooling) before it comes into the feed-forward layers. The comparison of model structures being used for designing the first DiffPool layer is shown in Figure 6.

Note that the size of the CEM Booleanized string is determined by the hidden unit number of our GNN layers. In our case, the CEM Boolenized string has a length of 32, which potentially enables a maximum of $2^{32}$ different clusters. Based on our experiments, we observe that although most of the $2^{32}$ possible clusters do not exist in the actual clustering, the resulting cluster number is still a very large number, usually above 400. We thus propose to select the most dominant clusters based on the number of data points each cluster contains. We propose a threshold to be some ratio of the total data size. For example, if the threshold is chosen to be 0.01, then we consider clusters having data points more than 1% of the total data size to be dominant. The number of those dominant clusters is then chosen to be the cluster number of the DiffPool layer to be decided.

### 4.3 Investigating Different CEM Guidance Model

We are aware that the choice of threshold essentially controls the suggested number of DiffPool clusters guided by the CEM clustering, hence we conduct an ablation study on the threshold selection. Table 3 shows the CEM guidance suggested cluster numbers for the first DiffPool layer under different threshold choices in $\{0.01, 0.02, 0.03, 0.04, 0.05\}$. The average and the standard deviation of cluster numbers shown in the table are calculated across 50 different seeds. The Vanilla GNN model used for CEM guidance has accuracy to be $79.13 \pm 2.01\%$.

Since we're constructing a three-DiffPool-layer model, we do not want the DiffPool clusters to be too few in the first DiffPool layer, hence we rule out the choice of threshold 0.04 and 0.05. We further

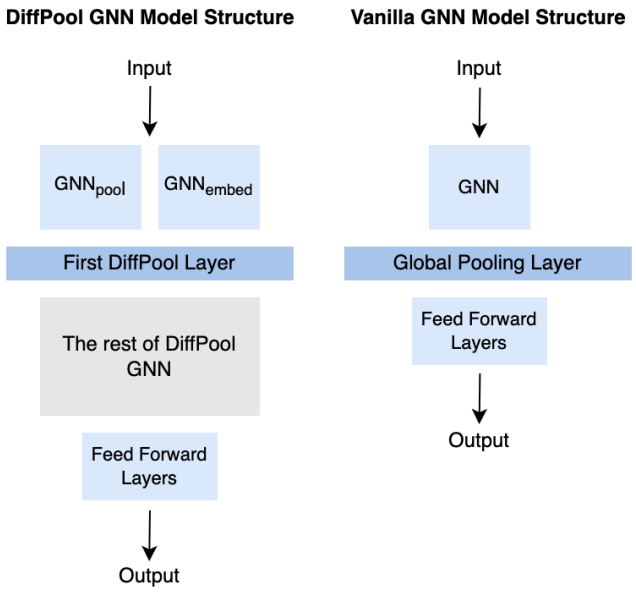



**Figure 6: Vanilla GNN Model Structure for CEM Guidance.** The DiffPool GNN model structure to be decided is shown on the left. We want to decide the number of new nodes for the first DiffPool layer (we opaque the structure of the rest of the model at this point). The Vanilla GNN model structure for CEM guidance is shown on the right. The Vanilla GNN has one GNN block, having the same configuration as the GNN blocks in the DiffPool model (i.e., 3 GCN layers in our case), one global pooling layer, and one feed-forward block which is also consistent with the one in the DiffPool model.

**Table 3:** CEM Guidance Suggested First DiffPool Cluster Number

| Threshold | Number of DiffPool Clusters |
|---|---|
| 0.01 | $22.04 \pm 2.87$ |
| 0.02 | $11.25 \pm 1.41$ |
| 0.03 | $6.56 \pm 1.56$ |
| 0.04 | $3.96 \pm 1.24$ |
| 0.05 | $2.81 \pm 0.96$ |

study the accuracy and interpretability of DiffPool models guided by CEM threshold 0.01, 0.02, and 0.03. Based on our previous experiments, we observe that the actual DiffPool concept numbers are usually smaller than the given DiffPool cluster numbers. We thus choose the cluster number to be slightly bigger than the suggested cluster number guided by CEM. We choose 24, 12, and 8 to be the corresponding DiffPool cluster numbers under threshold 0.01, 0.02, and 0.03.

In terms of the choice of cluster numbers for the second and third DiffPool layers, we fix the number of the third DiffPool clusters to be 2 because we want to coarsen the graphs to the largest extent without compressing everything into one single point. For the choice of the second DiffPool cluster number, one may follow the same procedure as we find the first DiffPool cluster number using CEM guidance. Namely, we may construct another GNN model with the same structure of the DiffPool GNN model before the second DiffPool layer and replace the second DiffPool layer with a global pooling layer. Similarly, we can perform CEM clustering on the latent space before the global pooling layer and then find the most dominant CEM clusters to decide the cluster number for the second DiffPool layer. Alternatively, we may choose the second DiffPool cluster number manually in-between the first and third DiffPool cluster numbers based on to what extent we expect the model to coarsen the graphs. In our study, we choose the model structures 24-8-2, 12-4-2, and 8-4-2 for the cluster numbers of the three DiffPool layers. Table 4 shows their accuracies and node concept completeness scores.

**Table 4:** Concept Completeness Scores for Different CEM Guidance Model

| Threshold | DiffPool Cluster Numbers | Accuracy (%) | Concept Completeness Score |
|---|---|---|---|
| 0.01 | 24-8-2 | 79.77±1.06 | **0.684** |
| 0.02 | 12-4-2 | 77.01±2.88 | 0.570 |
| 0.03 | 8-4-2 | 79.08±1.92 | 0.626 |

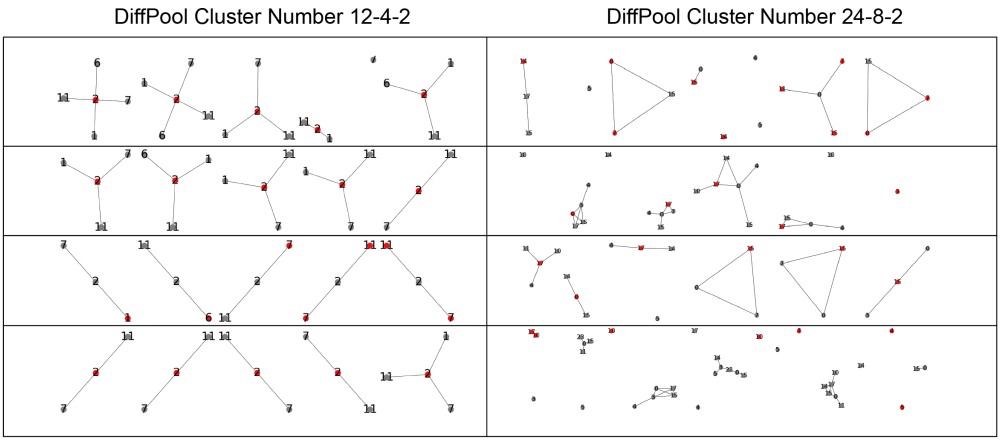



**Figure 7: Concept Visualization of the Second DiffPool Layer of CEM-Guided Models.** The human-interpretability of second layer DiffPool concepts extracted by these two CEM-guided models is worse than the model with DiffPool clusters 8-4-2.

From Table 4, it is evident that the CEM-guided model with a threshold of 0.01 yields the highest node concept completeness score of 0.684. The second highest is attained with a threshold of 0.03, while a threshold of 0.02 results in the lowest score. Additionally, thresholds 0.01 and 0.03 lead to better accuracies. However, Figure 7 reveals shortcomings in hierarchical concept extraction by the model guided by the 0.01 CEM threshold. Firstly, lower-level concepts that get pooled together are very different, particularly in the first cluster, where the pooled concepts vary considerably. Secondly, inconsistent structures exist within clusters, and each higher-level concept cluster exhibits structural disparities. For example, the third cluster contains both triangular and linear arrangements, suggesting inconsistency. In contrast, the model guided with the 0.02 CEM threshold shows some human-understandable patterns in hierarchical concept extraction, though the distinction between concept clusters is vague; clusters one and two appear similar, as do clusters three and four, indicating a deficiency in hierarchical concept delineation. We therefore conclude that the model guided by CEM threshold 0.03 demonstrates superior performance in extracting hierarchical concepts, balancing both completeness and distinguishable structure across clusters. This leads to our model choice of DiffPool cluster numbers 8-4-2

## 5 Discussion

**Quantifying Hierarchical Concept Quality.** In this project, we've explored the quality of hierarchical concepts extracted by analyzing the top five features nearest to the center of each DiffPool cluster, and displaying concept representations via graph visualization. The assessment of concept quality is based on observing the similarity of lower-level concepts within a cluster that contribute to a new higher-level concept, as well as the resemblance of neighborhood structures among these new higher-level concepts. Determining how to quantify the quality of hierarchical concepts poses a question for future research. Future works may include developing metrics or methods that can systematically evaluate the coherence and structural integrity of hierarchical concepts, potentially leading to deeper insights into the interpretability and effectiveness of the DiffPool model in extracting meaningful patterns from complex graphs.

**Improving Interpretability by Emphasizing Locality in Pooling.** DiffPool has the capacity to pool together nodes that are spatially distant, which may occasionally reduce the model's interpretability. For instance, as depicted in Figure 3, cluster 5 pools together amino groups(NH2) and hydroxyl groups(OH), despite their separation. To enhance interpretability, it would be preferable for such components to be pooled into different clusters. In this context, the adoption of a pooling method that prioritizes local structures becomes crucial. Although the GNN blocks in DiffPool learn from both adjacency matrix and node features, the generated assignment matrix fails to consider local connections. We suggest that a local pooling technique would place more emphasis on the adjacency matrix, ensuring that only neighboring structures are pooled together. This approach could significantly improve the model's ability to extract meaningful and interpretable hierarchical concepts.

## 6 Conclusion

This project aims at improving concept extraction for graph classification models. Specifically, we leverage the hierarchical pooling nature of DiffPool and propose a hierarchical concept extraction strategy by performing concept extraction on the latent space before and after each DiffPool layer. We evaluate our extracted node concepts via concept completeness score and higher-level concepts via concept representation. Our experiment results indicate that DiffPool is interpretable. However, the DiffPool model needs to be carefully designed in order to provide human-understandable explanations. We thus propose a CEM-guided model construction approach for deciding the number of clusters used in the DiffPool layer. Namely, we find the most dominant clusters from the CEM clustering and use the number of these dominant clusters to suggest the DiffPool cluster number. We also propose a future perspective for what an interpretable pooling technique needs, shedding light on how to improve concept-based explanations in a hierarchical manner. Namely, structural information shall be emphasized more together when considering feature information to ensure locality during node pooling. Additionally, developing methods for the quantitative evaluation of extracted concepts gives another direction for future research.

## Acknowledgements

We would like to express our deepest gratitude to Lucie Charlotte Magister for proposing the initial idea for this project and her huge support and guidance throughout the project. We would also like to thank Dr. Petar Veličković and Prof. Pietro Liò for delivering the fantastic lectures and designing the Geometric Deep Learning course. Their expertise and guidance have been the basis of our work.

## References

[1] Lucie Charlotte Magister, Dmitry Kazhdan, Vikash Singh, and Pietro Liò. GCExplainer: Human-in-the-loop concept-based explanations for graph neural networks, 2021. 1, 2, 3, 4

[2] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling, 2019. 1, 2

[3] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[4] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018.

[5] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks, 2019. 2

[6] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network, 2020.

[7] Minh N. Vu and My T. Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks, 2020. 2

[8] Jonas Jürß, Lucie Charlotte Magister, Pietro Barbiero, Pietro Liò, and Nikola Simidjievski. Everybody needs a little help: Explaining graphs via hierarchical concepts, 2023. 2

[9] Lucie Charlotte Magister, Pietro Barbiero, Dmitry Kazhdan, Federico Siciliano, Gabriele Ciravegna, Fabrizio Silvestri, Mateja Jamnik, and Pietro Lio. Encoding concepts in graph neural networks, 2022. 2, 3

[10] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations, 2019. 3, 5

[11] Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of medicinal chemistry*, 48 1:312–20, 2005. URL https://api.semanticscholar.org/CorpusID:21873061. 4

[12] Chih-Kuan Yeh, Been Kim, Sercan O. Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks, 2022. 6

## A Comparison of Different Model Structures

Figure 8 shows the accuracy comparison of different model structures (i.e., different choices of DiffPool layer numbers and cluster numbers in each DiffPool layer) discussed in this report. Specifically, our selected model has DiffPool cluster numbers 8-4-2, comparison with model 8-4 is discussed in Section 4.1, and comparison with models 12-4-2 and 24-8-2 is discussed in Section 4.3.

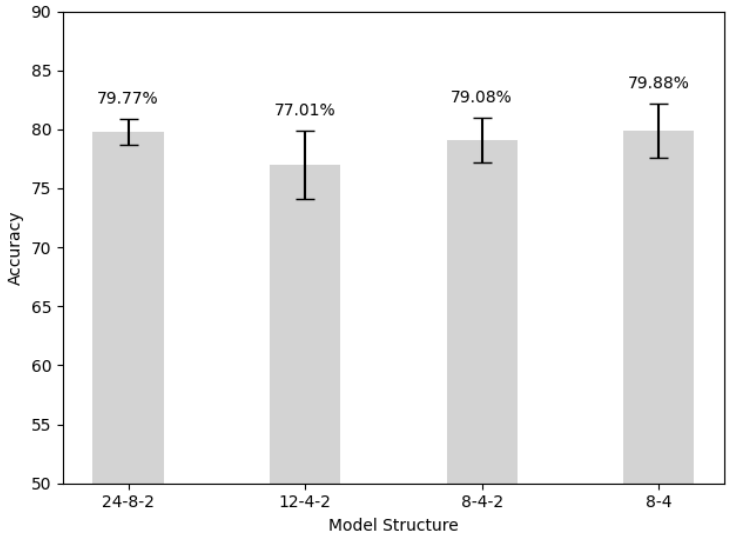



**Figure 8: Accuracy of Different Model Structures.** Accuracy of different model structures are shown with error bars calculated from the standard deviation across 10 random seeds. The models are named from the cluster numbers of each DiffPool layer in the model.

## B Comparison of Different Node Concept Extraction

As is discussed in Section 3, the node concepts are extracted via $k$-Means clustering on the latent space before the first DiffPool layer. In our selected model and all the experiments presented in this report, we choose $k = 30$, i.e., we have 30 node concepts. The node concept is on one hand evaluated via the concept completeness score as discussed in Section 3.3, on the other hand, we also consider how the number of node concepts influence the overall hierarchical concept extraction. We here discuss our choice by comparing with other possible node concept numbers via both the concept completeness score and concept representations. Table 5 shows the concept completeness scores for the node concepts under different node concept numbers.

As is shown in the table, the three choices are comparable and do not differ too much, and node concept number 50 appears to have a slightly higher concept completeness score compared with the other two. Given that the three scores are very close to each other, we claim that concept completeness score fails to serve as the selection criterion for node concept number in this case. We hence further analyze the concept representations.

From the graph visualization in Figure 9, we observe that the first DiffPool layer is also able to capture human-understandable characteristics from the 50 node concepts. Nonetheless, there's no strong

**Table 5:** Concept Completeness Score of Different Node Concept Extraction

| Number of Node Concepts | Concept Completeness Score |
|---|---|
| 30 | 0.626±0.023 |
| 40 | 0.625±0.020 |
| 50 | 0.635±0.019 |

evidence that the concept representations extracted under 50 node concepts outperform those under 30 node concepts. Therefore, we choose 30 node concepts both for less computation complexity during the $k$-Means clustering and for simpler hierarchical concept understanding.

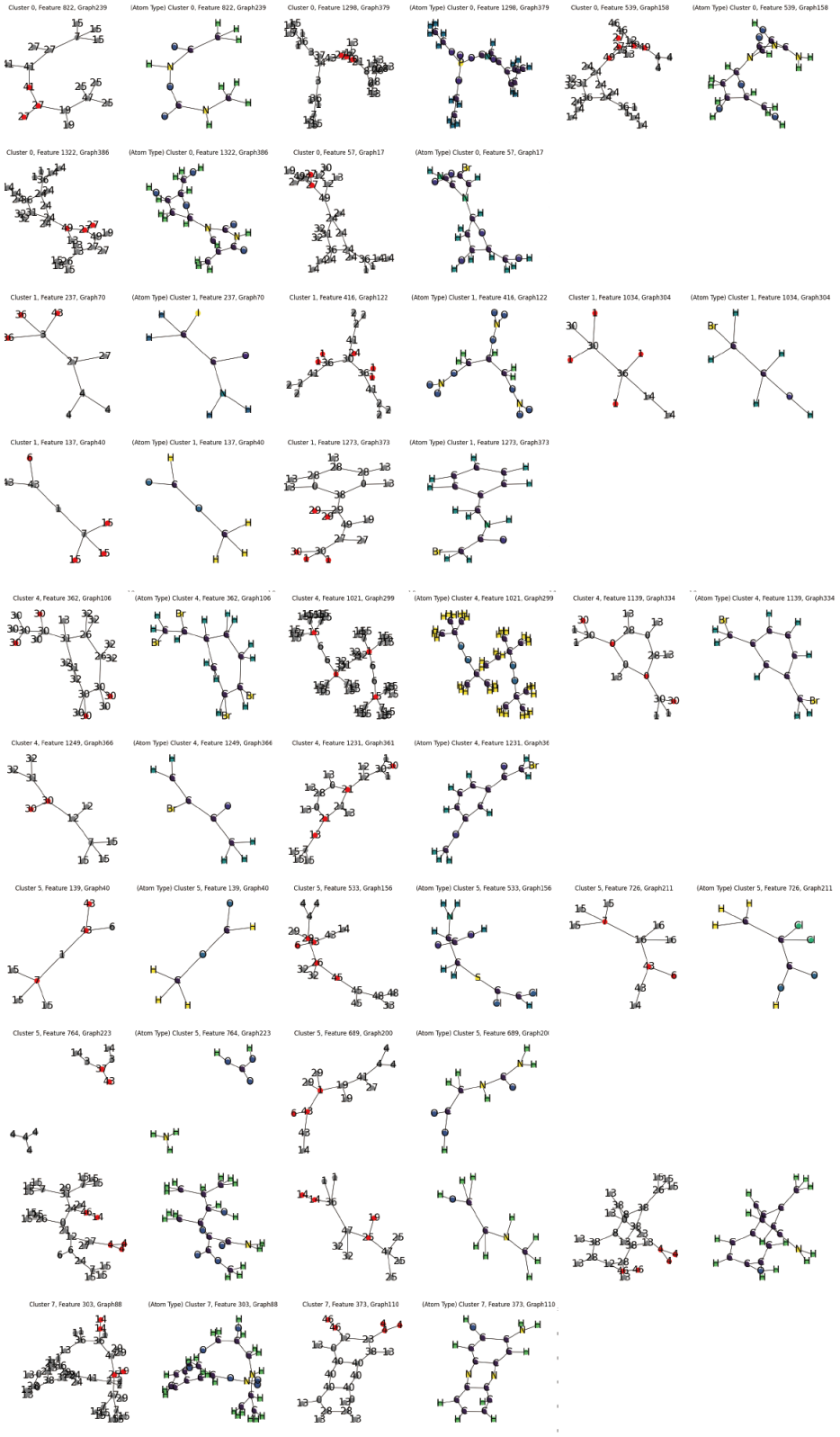


**Figure 9: Concept Visualization of the First DiffPool Layer with 50 Node Concepts.** The higher-level concepts within each cluster appear similar when 30 node concepts are assigned.