**Graph Clustering with Graph Neural Networks (1)**

The graph clustering source introduces Deep Modularity Networks (DMoN), a novel approach designed to overcome the limitations of existing GNN pooling techniques for graph clustering. DMoN distinguishes itself by directly optimizing a clustering objective within the GNN framework. This objective is rooted in the concept of modularity, a metric that quantifies the strength of a graph's division into clusters. A higher modularity score indicates a better partitioning, where nodes within clusters are more densely connected than nodes across different clusters. To adapt the modularity measure for GNNs, DMoN employs a differentiable formulation of the objective, allowing it to be integrated seamlessly into the end-to-end training process of the GNN. This ensures that both the node representations and cluster assignments are learned jointly, leading to more meaningful clusters than methods that apply clustering as a separate step after GNN training.

A key challenge in graph clustering is cluster collapse, where all nodes are assigned to a single cluster, essentially rendering the clustering meaningless. DMoN addresses this problem by incorporating a null model into its objective function. The null model represents a baseline expectation of the graph's connectivity pattern, and DMoN seeks to find clusters that deviate significantly from this baseline. By comparing against the null model, DMoN avoids creating trivial clusters that simply reflect the overall density of the graph and instead focuses on identifying groups of nodes with genuinely stronger connections than expected by chance. To further enhance cluster diversity, DMoN employs a collapse regularization technique. This technique encourages the formation of multiple clusters by penalizing solutions where the cluster sizes are too imbalanced. This regularization encourages the algorithm to explore a wider range of possible cluster assignments, leading to a more comprehensive and informative partitioning of the graph.

**Hierarchical Graph Explainability for Molecular Bioactivity Prediction (2)**

The graph hierarchical explainability aspect, as described in the sources, is a technique for gaining a deeper understanding of how graph neural networks (GNNs) predict molecular bioactivity. It is inspired by the Grad-CAM method used in image classification and extends it to the realm of molecular graphs. The core idea is to apply a hierarchy of explainers, each focusing on a different level of structural detail within the molecule, and then combine their insights to form a comprehensive explanation map.

Unlike traditional Grad-CAM, which analyzes only the final layer of a network, the hierarchical approach employs Grad-CAM explainers at different layers within the GNN. Each explainer focuses on a specific neighborhood size around each atom, capturing information at varying levels of structural complexity:

Layer 1 Explainer: Examines the immediate 1-hop neighborhood of each atom, essentially looking at direct bond partners. This helps identify atoms that are individually crucial for bioactivity.

Layer 3 Explainer: Analyzes the 3-hop neighborhood, encompassing small functional groups like rings or short chains. This reveals the contribution of these substructures to the overall bioactivity.

Layer 7 Explainer: Considers the entire molecule as the neighborhood, providing insights into the global structural patterns and their influence on bioactivity. This layer is analogous to the typical application of Grad-CAM in image classification.

The Grad-CAM values from each layer's explainer are summed for each atom to create a combined explanation score (LGrad-CAM). This score reflects the overall importance of an atom, considering its contribution at different structural levels.

**Believe the HiPe: Hierarchical perturbation for fast, robust, and model-agnostic saliency mapping (3)**

HiPe is a model-agnostic saliency mapping technique used to understand the predictions of AI models, particularly in image classification. The hierarchical aspect of HiPe stems from its approach of iteratively identifying salient regions within an image at increasing levels of resolution. HiPe starts by perturbing large, overlapping regions of the input image and observing the resulting change in the model's output. Regions that cause a significant drop in the confidence of the target class are deemed important and are further subdivided into smaller regions for subsequent perturbation. This process repeats, with HiPe focusing on increasingly smaller, more refined regions that contribute most significantly to the model's prediction. This hierarchical refinement allows HiPe to capture salient features at various scales, from large objects to fine-grained details, without requiring prior knowledge of the model's architecture or the size of the relevant features. As regions deemed less important are progressively disregarded, the computational cost is significantly reduced compared to other perturbation-based methods like RISE, which perturb the entire image with numerous random masks. Examples provided in the sources demonstrate HiPe's effectiveness in highlighting relevant regions for classification, such as identifying a baseball glove or a fork in images. Additionally, the sources mention that HiPe is data-agnostic and can be applied to other types of data beyond images, such as time-series data, video data, or graph data, by adapting the spatial relationship analysis.

**Everybody Needs a Little Help: Explaining Graphs Via Hierarchical Concepts (4)**

HELP, a novel graph learning method, distinguishes itself through its hierarchical concept extraction ability. Instead of relying solely on concepts from the final GNN layer, HELP iteratively pools the input graph into coarser representations, revealing how concepts from earlier layers combine to form new concepts in subsequent layers. Imagine this as looking at a molecule, first identifying individual atoms, then recognizing functional groups formed by these atoms, and finally understanding how these groups interact to determine the molecule's overall properties. This hierarchical pooling is achieved through a process of clustering node embeddings and merging connected components that belong to the same cluster.

At each pooling step, HELP applies multiple GNN layers and then clusters the resulting node embeddings. Connected components of nodes mapped to the same cluster are then merged, with the new node's embedding calculated as the average of the merged nodes' embeddings4. This approach allows HELP to identify relevant substructures at different levels of the graph's hierarchy, leading to more nuanced and insightful explanations compared to methods focusing only on the final layer3. For instance, in a social network, HELP might first identify individual users, then group them into communities based on their interactions, and finally understand how these communities are interconnected3. This hierarchical approach enables HELP to capture the complex interplay of concepts across different layers of a GNN, moving beyond simplistic explanations towards a more comprehensive understanding of the model's decision-making process.

**Explaining Temporal Graph Models through an Explorer-Navigator Framework (5)**  
  
The paper "EXPLAINING TEMPORAL GRAPH MODELS THROUGH AN EXPLORER-NAVIGATOR FRAMEWORK" introduces T-GNNExplainer, a novel method for explaining predictions made by temporal graph models. Recognizing the limitations of existing static graph explainers in capturing temporal dependencies, T-GNNExplainer utilizes a unique explorer-navigator framework to identify influential historical events that contribute to a specific prediction. The explorer employs Monte Carlo Tree Search (MCTS) to efficiently search for the most relevant subset of events, while the navigator, a pre-trained neural network, guides the search process by learning the correlations between events. This learned guidance helps reduce the search space and improve both the speed and accuracy of explanations. The paper demonstrates T-GNNExplainer's superior performance compared to baseline methods on synthetic and real-world datasets, showcasing its ability to identify concise and accurate explanations for temporal graph model predictions.

**Explainability Methods for Graph Convolutional Neural Networks (6)**

The paper "Explainability Methods for Graph Convolutional Neural Networks" explores adapting popular explainability methods, originally developed for Convolutional Neural Networks (CNNs), to the realm of Graph Convolutional Neural Networks (GCNNs). The authors focus on three prominent methods: contrastive gradient-based saliency maps, Class Activation Mapping (CAM), and Excitation Backpropagation (EB), including variants like Grad-CAM and contrastive EB. These methods are extended to identify and highlight important nodes, edges, or subgraphs within a graph that contribute significantly to a GCNN's prediction. The paper showcases the application of these techniques to classification tasks on visual scene graphs and molecular graphs. To evaluate the effectiveness of each method, the authors define quantitative metrics: fidelity (measuring the impact of occlusions), contrastivity (assessing the distinction between different classes), and sparsity (evaluating the conciseness of the explanations). By analyzing these metrics, the authors provide insights into the strengths and weaknesses of each explainability method, enabling researchers to select the most appropriate technique for a given application and gain a better understanding of GCNNs' decision-making process.

Reference:

[1] A. Tsitsulin, J. Palowitch, B. Perozzi, and E. Müller, "Graph clustering with graph neural networks," J. Mach. Learn. Res., vol. 24, pp. 1-21, May 2023.

[2] P. Sortino, S. Contino, U. Perricone, and R. Pirrone, "Unveiling Molecular Moieties through Hierarchical Graph Explainability," presented on May 9, 2024.

[3] J. Cooper, O. Arandjelovic, and D. J. Harrison, "Believe the HiPe: Hierarchical perturbation for fast, robust, and model-agnostic saliency mapping," Pattern Recognition, vol. 131, p. 108914, Apr. 2022.

[4] Everybody needs a little help: Explaining graphs via hierarchical concepts," unpublished manuscript, under double-blind review, 2024.

[5] W. Xia, M. Lai, C. Shan, Y. Zhang, X. Dai, X. Li, and D. Li, "Explaining Temporal Graph Models through an Explorer-Navigator Framework," in Proc. Int. Conf. Learn. Representations (ICLR), May 2023.

[6] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability Methods for Graph Convolutional Neural Networks," n Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2023.