Self-Interpretable Graph Learning with Sufficient and Necessary Explanations

Anonymous submission

A. Soft Augmentation on $G_{S_{att}}$

During the training phase, the explanation G_S takes the form of an attentive subgraph $G_{S_{att}}$, wherein the edge weight is assigned in accordance with M, based on the computation graph G_C. Consequently, performing augmentations on G_S is the same as performing them on $G_{S_{att}}$, that is, perturbing the saliency edge mask M. We increase the edge weight of sampled positive edges, equivalent to adding them into G_S, and reduce the edge weight of sampled negative edges, equivalent to removing them from G_S.

As detailed in Algorithm 1, we utilize M as input to generate the perturbed edge maps M^+ and M^-. Initially, we compute the average importance scores m^+ and m^- derived from M (lines 4 to 6). The score m^+ represents the average score of the complement of explanation (i.e., $G_C \backslash G_S$), while m^- is the average edge importance score of G_S. Then we compute the distance coefficients δ^+ and δ^- for refieing M (line 7). Next, we sample r^+ edges using $M \odot \delta^+$, and replace their corresponding edge weight with m^+ to construct positive samples (lines 8 to 13), which is identical to inserting these sampled edges into G_S. Similarly, we sample r^- edges using $M \odot \delta^-$, and replace their corresponding edge weight with m^- to construct negative samples (lines 14 to 19), which is the same as removing these sampled edges from G_S. The produced M^+ and M^- are then utilized as attentive subgraphs $G^+_{S_{att}} = G_C \odot M^+$ and $G^-_{S_{att}} = G_C \odot M^-$ in subsequent training processes.

Algorithm 1: Soft Augmentation.

Input: Saliency map $M \in \mathbb{R}^{|\mathcal{E}_C|}$, # of positive and negative samples $\{n^+, n^-\}$, explanation sample ratio k.
Output: Augmented positive and negative samples $\{M^+, M^-\}$.

1: # of total edges $N_C \leftarrow |\mathcal{E}_C|$
2: # of edges in the explanation $N_S \leftarrow \lfloor k|\mathcal{E}_C| \rfloor$
3: $M^+ \leftarrow \varnothing$ and $M^- \leftarrow \varnothing$
4: Rank M with descending order
5: $m^+ \leftarrow \text{Average}(M[1, \cdots, N_S])$
6: $m^- \leftarrow \text{Average}(M[N_{S+1}, \cdots, N_C])$
7: Compute distance coefficient δ^+ and δ^- with Eq. 1 and Eq. 2
8: **for** $i \in n^+$ **do**
9: Sample r^+ elements from $M[N_{S+1}, \cdots, N_C]$ using $M \odot \delta^+$, and they are indexed by idx_i^+
10: $M_i^+ \leftarrow \text{copy}(M)$
11: $M_i^+[idx_i^+] \leftarrow m^+$
12: Append M_i^+ into M^+
13: **end for**
14: **for** $j \in n^-$ **do**
15: Sample r^- elements from $M[1, \cdots, N_S]$ using $M \odot \delta^-$, and they are indexed by idx_j^-
16: $M_j^- \leftarrow \text{copy}(M)$
17: $M_j^-[idx_j^-] \leftarrow m^-$
18: Append M_j^- into M^-
19: **end for**
20: return $\{M^+, M^-\}$

B. Adapting SNEX-GNN to Heterogeneous Graphs

Definition 1 (Heterogeneous Graph) *A directed heterogeneous graph $G = (\mathcal{E}, \mathcal{V}, \mathcal{N}, \mathcal{R})$, where each node $v \in \mathcal{V}$ and each edge $e \in \mathcal{E}$ are associated with their type mapping functions $o(v) : \mathcal{V} \to \mathcal{N}$ and $\kappa(e) : \mathcal{E} \to \mathcal{R}$, respectively.*

Explanation Generator. We now consider adapting the explanation generator h to heterogeneous scenarios. We maintain different MLP-based map functions p_μ to project nodes of different types into a common dimension. We set the common dimension as the dimension of target node v in the 2nd layer, i.e., $d^{(2)}$. For the simplification of notations, we omit the parameter μ for p. For a directed edge (i, j), where node i is of type n_i and node j is of type n_j, we compute its importance score by:

- $m_{ij} = h_\theta[p_{n_i}(z_i^{(0)})||p_{n_j}(z_j^{(1)})||z_v^{(2)}]$, if (i, j) connects nodes in layer 0 and layer 1;
- $m_{ij} = h_\theta[p_{n_i}(z_i^{(1)})||p_{n_j}(z_j^{(2)})||z_v^{(2)}]$, if (i, j) connects nodes in layer 1 and layer 2;

where p_{n_i} is the mapping function for node type n_i and $||$ is the concatenation operation.

Augmentation. Patterns behind meta-path are believed to imply different and useful semantics (Lv et al. 2021). For example, "$author \leftrightarrow paper \leftrightarrow author$" meta-path defines the "coauthor" relationship. We take meta-path pattern information into consideration to help construct hard positive

and negative samples.

Definition 2 (Meta Path) *A meta-path is a path with a predefined (node or edge) types pattern, i.e.,* $\mathcal{P}_t = n_1 \xrightarrow{r_1} n_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l} n_{l+1}$*, where* $n_i \in \mathcal{N}$ *and* $r_i \in \mathcal{R}$.

Recall that we add edges to and remove edges from the explanation G_S for guiding h to generate explanations toward sufficiency and necessity directions, respectively. Intuitively, *paths that instantiate meta-paths are more informative than those that do not*. To construct positive samples, we prioritize the edges that, when added to G_S, will instantiate a predefined meta-path. These samples are considered hard positive samples since they introduce more information and make it harder to optimize h toward the sufficiency direction. Similarly, to construct negative samples, we prioritize the edges that are not present in the meta-path instances. These samples are considered hard negative samples since the corrupted edges are not informative and make it harder to optimize h toward the necessity direction.

Formally, we pre-define a set of meaningful meta-paths $\mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_T\}$. We first record all simple paths within the computation graph G_C and calculate the importance scores of the paths that satisfy the meta-paths in $\mathcal{P}$. We then sum up these scores to obtain $s \in \mathbb{R}$. We first omit the type information and construct positive and negative samples $\{G^+_{S_{att}}, G^-_{S_{att}}\}$ as the way we do for homogeneous graphs. Then we introduce type information into the augmented samples and record their sum of importance scores of paths satisfying meta-paths in $\mathcal{P}$ as $s^+ \in \mathbb{R}^{n^+}$ and $s^- \in \mathbb{R}^{n^-}$. We then introduce **type coefficients** β^+ and β^- to reweight the augmented samples:

$$\beta^+ = \{(s^+_1 - s), \cdots, (s^+_{n^+} - s)\} \in \mathbb{R}^{n^+}, \tag{1}$$

$$\beta^- = \{(s - s^-_1), \cdots, (s - s^-_{n^-})\} \in \mathbb{R}^{n^-}. \tag{2}$$

Higher β^+ indicates the positive sample introduces rich semantic information into G_S, and higher β^- indicates the negative sample corrupts less semantic information from G_S. Samples with higher **type coefficients** are considered harder samples.

C. Implementation Details

We run all the experiments on a server with Intel(R) Xeon(R) Silver 4110 CPU, 128GB Memory, and an Nvidia GeForce RTX 2080 Ti GPU (12GB Memory). The hyperparameters for the training of SNEX-GNN as listed as follows:

- Temperature for reparameterization trick: the reparameterization trick is leveraged to approximate a discrete saliency map from the original soft mask. Following previous works (Miao, Liu, and Li 2022), we set the temperature to 1;
- Distance Coefficients: we set the non-negative constant α of distance coefficients to 0.15;
- Number of augmented positive and negative samples: we set $n^+ = n^- = 10$;

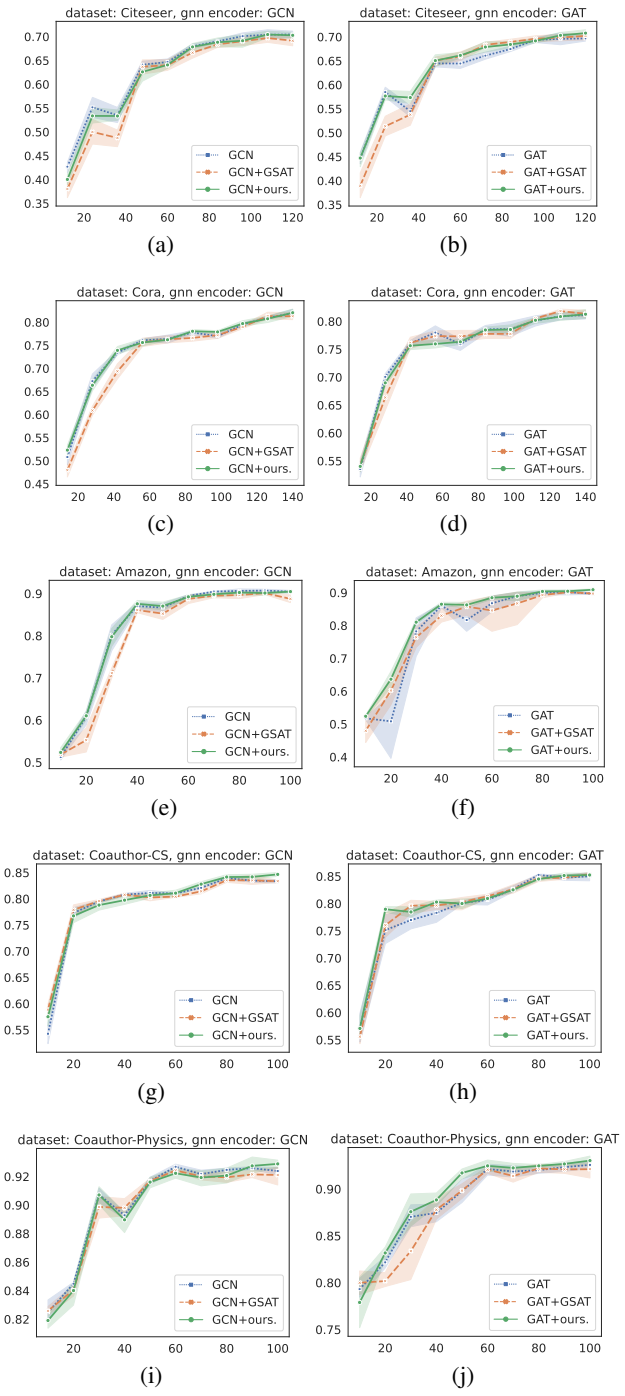

Figure 1: The classification accuracy (y-axis) of the model as the number of labeled nodes in the input graph (x-axis) changes.

Table 1: Explainability performance and standard deviations (%).

	Pubmed		Amazon		Coauthor-CS		Coauthor-Physics	
	$fid_+ \uparrow$	$fid_- \downarrow$	$fid_+ \uparrow$	$fid_- \downarrow$	$fid_+ \uparrow$	$fid_- \downarrow$	$fid_+ \uparrow$	$fid_- \downarrow$
GCN								
+ GNNExplainer	37.32±8.4	1.98±1.2	25.5±1.4	2.59±1.3	8.38±0.4	8.62±0.3	6.92±0.2	5.84±0.3
+ PGExplainer	58.91±3.2	0.23±0.2	19.67±1.9	9.82±5.0	9.39±5.8	14.08±9.6	20.15±2.6	7.58±6.7
+ ReFine	56.08±2.4	0.00±0.0	84.23±8.8	1.78±3.2	94.20±4.2	1.92±0.1	71.65±9.9	5.11±2.1
+ GSAT	51.17±13.7	1.77±2.0	9.82±5.2	3.55±1.1	1.98±1.6	2.82±0.5	4.29±2.4	2.21±0.8
+ CAL	61.52±11.6	6.70±1.5	90.98±5.0	4.28±0.5	87.62±1.6	3.80±1.0	74.15±3.4	2.67±0.4
+ SNEX-GNN	64.26±8.8	0.00±0.0	85.28±5.6	0.86±1.1	89.85±8.2	7.20±5.8	77.34±5.6	1.93±9.6
GAT								
+ GNNExplainer	21.60±11.8	1.14±1.9	11.25±0.6	2.39±2.0	9.00±0.2	1.99±0.5	2.39±0.2	0.97±0.3
+PGExplainer	55.33±3.2	0.28±0.2	22.16±0.2	6.44±6.1	3.78±0.3	8.31±4.9	7.58±1.1	8.61±6.1
+ ReFine	55.46±3.2	0.00±0.0	69.29±1.5	0.00±0.0	83.17±4.8	0.00±0.0	82.55±4.4	0.00±0.0
+ GSAT	56.03±6.2	2.28±1.9	13.18±2.5	1.36±0.8	2.47±1.3	2.37±0.5	1.26±2.0	0.71±0.3
+ CAL	60.34±14.5	3.02±1.2	80.31±11.0	5.67±3.5	81.98±4.9	5.62±1.7	78.56±16.0	3.03±1.2
+ SNEX-GNN	58.46±1.5	0.00±0.0	74.34±6.0	0.13±0.3	84.10±4.8	0.00±0.0	87.59±4.3	0.00±0.0

- Explanation sampling ratio: the hyperparameter $k = |\mathcal{E}_S|/|\mathcal{E}_C|$ is used to control the proportion of explainable edges sampled from the original graph. The value of k is set to 0.3 at the beginning of training, and gradually decreased as training progresses, ultimately reaching 0.1. To provide human-understandable explanations after training, we control the number of edges involved in the explanations, e.g., $|\mathcal{E}_S| = 5$.
- Perturbation ratio: the perturbation ratio r^+ and r^- are used to control the number of sampled edges when constructing positive and negative samples. We set $r^+ = r^- = 0.05$, that is, we perturb $\lfloor r^+|\mathcal{E}_C| \rfloor$ and $\lfloor r^-|\mathcal{E}_C| \rfloor$ edges when constructing positive and negative samples;
- Trade-off hyperparameter of contrastive loss: we set $\gamma = 0.01$;
- Temperature for contrastive loss: we set the temperature hyperparameter $\tau = 0.1$ to improve the robustness of our model.

D. Extensive Experiments

Robustness to Few-label Issue

To further evaluate the robustness of SNEX-GNN when encountering the situation that only few nodes are with labels, we conduct extensive experiments. As illustrated in Fig. 1, we gradually reduce the number of labeled nodes and observe the prediction accuracy of baselines including the base GNNs, GSAT and SNEX-GNN. We observed that as the number of labels decreases, the prediction accuracy of GNNs tends to decline to some extent. In most cases, SNEX-GNN approximates or outperforms base GNNs in most cases. This is because data augmentation makes SNEX-GNN more robust to the few-label issue. On the other hand, GSAT, which relies solely on supervised loss optimization, fails when the number of labels drastically decreases.

Evaluations on Explainability Fidelity

We evaluate the explainability fidelity of all base explainers in all datasets. Table 1 presents comprehensive experimental results on explainability fidelity in datasets except for Citeseer and Cora which are already accessible in the main paper. We observe that SNEX-GNN is able to achieve stable explainability metrics and perform optimally under most conditions. Explanations generated by GSAT are more likely to demonstrate the sufficiency on larger datasets such as Amazon, Coauthor-Physics, and Coauthor. It can achieve relatively better scores on the fid_- metric but almost collapses on the fid_+ metric. The same situation occurs with GNNExplainer and PGExplainer. This suggests that finding necessary explanations becomes more challenging on graphs with more complex local relationships. On the other hand, CAL is able to achieve relatively higher fid_+ scores. However, when considering Table 2 in conjunction with the main paper, it becomes evident that such necessary explanations have limited improvement on model predictions and may even have negative effects. This is because the explanations they generated contain very limited information, which is insufficient for GNNs to make correct predictions. SNEX-GNN addresses this issue by optimizing the contrastive loss to ensure the quality of generated explanations from both sufficiency and necessity directions. This ensures that the information used by GNNs is sufficient and free from harmful noise.

References

Lv, Q.; Ding, M.; Liu, Q.; Chen, Y.; Feng, W.; He, S.; Zhou, C.; Jiang, J.; Dong, Y.; and Tang, J. 2021. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 1150–1160.

Miao, S.; Liu, M.; and Li, P. 2022. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*, 15524–15543. PMLR.