A Proof of Lemma 2

Proof. We prove by contradiction. Suppose Lemma 2 is false, then either of the following shall hold:

i) There exists two variables $\{X_i, X_j\} \subset \mathbf{X}$, which are not equally contribute to $g(\mathbf{X})$ at (x_i, x_j) with respect to $(0, 0)$;

ii) There are two variables $X_i, X_j \subset \mathbf{X}$ that cannot have $X_i = x_i$ and $X_j = 0$ simultaneously, or $X_i = 0$ and $X_j = x_j$ simultaneously.

We consider contradiction to each of the above statements separately.

i) According to Definition 1, if there exists two variables $\{X_i, X_j\} \subset \mathbf{X}$, which are not equally contribute to $g(\mathbf{X})$ at (x_i, x_j) with respect to $(0, 0)$, we should have at least one assignment of $\mathbf{X}$ except for X_i and X_j such that

$$g_{X_i=x_i,X_j=0}(\mathbf{X}\backslash\{X_i, X_j\}) \neq g_{X_i=0,X_j=x_j}(\mathbf{X}\backslash\{X_i, X_j\}).$$

However, since

$$g_{X_i=x_i,E_j=0}(\mathbf{X}\backslash\{X_i, X_j\}) = g_{X_i=0,X_j=x_j}(\mathbf{X}\backslash\{X_i, X_j\}) = 0,$$

we have reached a contradiction to the first statement.

ii) As all the variables are uncorrelated, the value assigned to one variable has no impact on how we choose values for the other variables. Therefore, it is possible to have $X_i = x_i$ and $X_j = 0$ simultaneously or to have $X_i = 0$ and $X_j = x_j$ simultaneously for arbitrary two variables X_i and X_j. We have reached a contradiction to the second statement. □

B Proof of Lemma 4

Proof. To prove Lemma 4, we first prove the following Lemma.

Lemma B.1. *For a scalar product term z in the expansion form of a pretrained GNN $f(\cdot)$, when the number of nodes N is large, it is feasible to have both $\nu_i = x_i$ and $\nu_j = 0$ or both $\nu_i = 0$ and $\nu_j = x_j$ for all possible pairs ν_i, ν_j of variables in z, where x_i, x_j indicate the presence of variables ν_i, ν_j, respectively.*

Proof. We first show that in the scenario of a large number of nodes N, an arbitrary variable $P_{c,e}$ among the variables in z can take any value within its domain while keeping all other variables in z fixed to certain values.

We begin with defining notations. For a scalar product term z in the expansion of a pretrained GNN $f(\cdot)$, we let $\mathbf{U} = A^{(L)}_{\alpha_{L0},\alpha_{L1}} \dots A^{(1)}_{\alpha_{10},\alpha_{11}}$ denotes the factors involving the adjacency matrix A, $\mathbf{P} = P^{(1)}_{\alpha_{10},\beta_{11}} \dots P^{(L)}_{\alpha_{L0},\beta_{L1}} \cdot P^{(\mathrm{c}_1)}_{\alpha_{L0},\gamma_{11}} \dots P^{(\mathrm{c}_{(M-1)})}_{\alpha_{L0},\gamma_{(M-1)1}}$ denotes the factors involving the activation pattern P, and $\mathbb{C} = W^{(1)}_{\beta_{10},\beta_{11}} \dots W^{(L)}_{\beta_{L0},\beta_{L1}} \cdot W^{(\mathrm{c}_1)}_{\gamma_{10},\gamma_{11}} \dots W^{(\mathrm{c}_M)}_{\gamma_{M0},\gamma_{M1}}$ stands for "reduced to constant" denoting the product of the related parameters in $f(\cdot)$, such that each product term can be rewritten as $z = \mathbf{UP}X_{i,j}\mathbb{C}$.

The entries in an activation pattern are determined by the hidden representation before being passed to the activation function. Similar to Equation (9), the (c, e)-th entry of the hidden representation at the l-th layer before the activation function can be expressed by the sum of all the related scalar products as

$$h^{(l)\prime}_{c,e} = \sum^{\alpha,\beta,\rho} \left[\left(A^{(l)}_{c,\alpha_{l1}} W^{(l)}_{\beta_{l0},e} \prod_{m=1}^{l-1} P^{(m)}_{\alpha_{m0},\beta_{m1}} A^{(m)}_{\alpha_{m0},\alpha_{m1}} W^{(m)}_{\beta_{m0},\beta_{m1}} \right) X_{\rho_{m0},\rho_{m1}} \right].$$

When none of the variables in $h^{(l)\prime}_{c,e}$ are constrained, the mathematical range of $h^{(l)\prime}_{c,e}$ is $\mathbb{R}$. Let $c(h^{(l)\prime}_{c,e}, z)$ be the sum of scalar products involving the variables in z. If the range of $(h^{(l)\prime}_{c,e} - c(h^{(l)\prime}_{c,e}, z))$ is also $\mathbb{R}$ when none of the variables in it is constrained, then $P_{c,e}$ can take any value within its domain while keeping the variables in z fixed to some certain values. In other words, if there is at least one scalar product in $(h_{c,e}^{(l)\prime} - c(h_{c,e}^{(l)\prime}, z))$, then $P_{c,e}$ can take any value within its domain while holding the variables in z fixed to certain values.

The sum of scalar products involving $X_{i,j}$ is

$$c(h_{c,e}^{(l)\prime}, X_{i,j}) = \sum^{\alpha,\beta} \left[\left(A_{c,\alpha_{l1}}^{(l)} W_{\beta_{l0},e}^{(l)} \prod_{m=1}^{l-1} P_{\alpha_{m0},\beta_{m1}}^{(m)} A_{\alpha_{m0},\alpha_{m1}}^{(m)} W_{\beta_{m0},\beta_{m1}}^{(m)} \right) X_{i,j} \right].$$

The sum of scalar products involving an arbitrary variable $A_{a,b}$ in $\mathbf{U}$ is

$$c(h_{c,e}^{(l)\prime}, A_{a,b}) = \sum^{\alpha,\beta,\rho} \left[\left(A_{c,\alpha_{l1}}^{(l)} W_{\beta_{l0},e}^{(l)} \prod_{m=1}^{l-1} P_{\alpha_{m0},\beta_{m1}}^{(m)} A_{\alpha_{m0},\alpha_{m1}}^{(m)} W_{\beta_{m0},\beta_{m1}}^{(m)} \right) X_{\rho_{m0},\rho_{m1}} \right], \text{where}$$
$$\text{at least one of } \{A_{c,\alpha_{l1}}^{(l)}, A_{\alpha_{(l-1)0},\alpha_{(l-1)1}}^{(l-1)}, \dots, A_{\alpha_{10},\alpha_{11}}^{(1)}\} \text{ is } A_{a,b}.$$

The sum of scalar products involving an arbitrary variable $P_{g,h}$ in $\mathbf{P}$ is

$$c(h_{c,e}^{(l)\prime}, P_{g,h}) = \sum^{\alpha,\beta,\rho} \left[\left(A_{c,\alpha_{l1}}^{(l)} W_{\beta_{l0},e}^{(l)} \prod_{m=1}^{l-1} P_{\alpha_{m0},\beta_{m1}}^{(m)} A_{\alpha_{m0},\alpha_{m1}}^{(m)} W_{\beta_{m0},\beta_{m1}}^{(m)} \right) X_{\rho_{m0},\rho_{m1}} \right], \text{where}$$
$$\text{one of } \{P_{\alpha_{(l-1)0},\beta_{(l-1)1}}^{(l-1)}, \dots, P_{\alpha_{10},\beta_{11}}^{(1)}\} \text{ is } P_{g,h}.$$

Then the number of scalar products in $(h_{c,e}^{(l)\prime} - c(h_{c,e}^{(l)\prime}, z))$ is

$$|h_{c,e}^{(l)\prime} - c(h_{c,e}^{(l)\prime}, z)| \geq |h_{c,e}^{(l)\prime}| - |c(h_{c,e}^{(l)\prime}, X_{i,j})| - l \cdot |c(h_{c,e}^{(l)\prime}, A_{a,b})| - (l-1) \cdot |c(h_{c,e}^{(l)\prime}, P_{g,h})|,$$

where $|h_{c,e}^{(l)\prime}|, |c(h_{c,e}^{(l)\prime}, X_{i,j})|, |c(h_{c,e}^{(l)\prime}, A_{a,b})|, |c(h_{c,e}^{(l)\prime}, P_{g,h})|$ represents the number of scalar products in $h_{c,e}^{(l)\prime}, c(h_{c,e}^{(l)\prime}, X_{i,j}), c(h_{c,e}^{(l)\prime}, A_{a,b}), c(h_{c,e}^{(l)\prime}, P_{g,h})$ respectively. Hence if we can prove $|h_{c,e}^{(l)\prime}| - |c(h_{c,e}^{(l)\prime}, X_{i,j})| - l \cdot |c(h_{c,e}^{(l)\prime}, A_{a,b})| - (l-1) \cdot |c(h_{c,e}^{(l)\prime}, P_{g,h})| \geq 1$, then we will also have $|h_{c,e}^{(l)\prime} - c(h_{c,e}^{(l)\prime}, z)| \geq 1$ proved. That is, we should prove

$$\frac{|h_{c,e}^{(l)\prime}|}{|h_{c,e}^{(l)\prime}|} - \frac{|c(h_{c,e}^{(l)\prime}, X_{i,j})|}{|h_{c,e}^{(l)\prime}|} - l \cdot \frac{|c(h_{c,e}^{(l)\prime}, A_{a,b})|}{|h_{c,e}^{(l)\prime}|} - (l-1) \cdot \frac{|c(h_{c,e}^{(l)\prime}, P_{g,h})|}{|h_{c,e}^{(l)\prime}|} \geq \frac{1}{|h_{c,e}^{(l)\prime}|}.$$

Equivalently, we should prove

$$\frac{|c(h_{c,e}^{(l)\prime}, X_{i,j})|}{|h_{c,e}^{(l)\prime}|} + l \cdot \frac{|c(h_{c,e}^{(l)\prime}, A_{a,b})|}{|h_{c,e}^{(l)\prime}|} + (l-1) \cdot \frac{|c(h_{c,e}^{(l)\prime}, P_{g,h})|}{|h_{c,e}^{(l)\prime}|} \leq 1 - \frac{1}{|h_{c,e}^{(l)\prime}|}.$$

Note that the first term

$$\frac{|c(h_{c,e}^{(l)\prime}, X_{i,j})|}{|h_{c,e}^{(l)\prime}|} = \frac{1}{Nd}.$$

Since $d \geq 1$ is the feature dimension of X, we have $\lim_{N \to \infty} \frac{|c(h_{c,e}^{(l)\prime}, X_{i,j})|}{|h_{c,e}^{(l)\prime}|} = 0$.

Now consider the second term $l \cdot \frac{|c(h_{c,e}^{(l)\prime}, A_{a,b})|}{|h_{c,e}^{(l)\prime}|}$:

$$\begin{aligned}
l \cdot \frac{|c(h_{c,e}^{(l)\prime}, A_{a,b})|}{|h_{c,e}^{(l)\prime}|} &= l \cdot \frac{\binom{l}{1} N^{(l-1)} + \binom{l}{2} N^{(l-2)} + \cdots + \binom{l}{l} N^0}{N^{(l+1)}} \\
&= l \cdot \left(\frac{\binom{l}{1}}{N^2} + \frac{\binom{l}{2}}{N^3} + \cdots + \frac{\binom{l}{l}}{N^{(l+1)}} \right) \\
&= l \cdot \left(\frac{l!}{1!(l-1)! \cdot N^2} + \frac{l!}{2!(l-2)! \cdot N^3} + \cdots + \frac{l!}{l!(l-l)! \cdot N^{(l+1)}} \right) \\
&= \left(\frac{l}{1!N} \cdot \frac{l}{N} \right) + \left(\frac{l}{2!N} \cdot \frac{l}{N} \cdot \frac{l-1}{N} \right) + \cdots + \left(\frac{l}{l!N} \cdot \frac{l}{N} \cdot \frac{l-1}{N} \cdots \frac{1}{N} \right).
\end{aligned}$$

Since $N \gg l$, we have $\lim_{N\to\infty} \frac{l}{N} = 0$. Also, because $\frac{l}{N} > \frac{l-1}{N} > \cdots > \frac{1}{N} > \frac{1}{1!N} > \cdots > \frac{1}{l!N}$, we then have $\lim_{N\to\infty} l \cdot \frac{|c(h_{c,e}^{(l)\prime}, A_{a,b})|}{|h_{c,e}^{(l)\prime}|} = 0$.

Now we consider the third term $(l-1) \cdot \frac{|c(h_{c,e}^{(l)\prime}, P_{g,h})|}{|h_{c,e}^{(l)\prime}|}$:

$$(l-1) \cdot \frac{|c(h_{c,e}^{(l)\prime}, P_{g,h})|}{|h_{c,e}^{(l)\prime}|} = (l-1) \cdot \frac{1}{N^l d}$$

Since $N \gg l$ and $d \geq 1$, we have $\lim_{N\to\infty} (l-1) \cdot \frac{|c(h_{c,e}^{(l)\prime}, P_{g,h})|}{|h_{c,e}^{(l)\prime}|} = 0$.

For the terms on the right hand side of the inequality, since $\frac{1}{|h_{c,e}^{(l)\prime}|} \leq \frac{|c(h_{c,e}^{(l)\prime}, X_{i,j})|}{|h_{c,e}^{(l)\prime}|}$, we have $\lim_{N\to\infty} \frac{1}{|h_{c,e}^{(l)\prime}|} = 0$. Hence $\lim_{N\to\infty} 1 - \frac{1}{|h_{c,e}^{(l)\prime}|} = 1$.

Since $0 < 1$, we have prove that when N is large, $\frac{|c(h_{c,e}^{(l)\prime}, X_{i,j})|}{|h_{c,e}^{(l)\prime}|} + l \cdot \frac{|c(h_{c,e}^{(l)\prime}, A_{a,b})|}{|h_{c,e}^{(l)\prime}|} + (l-1) \cdot \frac{|c(h_{c,e}^{(l)\prime}, P_{g,h})|}{|h_{c,e}^{(l)\prime}|} \leq 1 - \frac{1}{|h_{c,e}^{(l)\prime}|}$. Therefore, in scenarios with a large number of nodes N, an arbitrary variable $P_{c,e}$ in $\mathbf{P}$ can take any value within its domain while keeping all other variables in z fixed to certain values.

If the data is properly preprocessed, a feature $X_{i,j}$ and the unique entries in A should be uncorrelated with each other. Also, from the above proof we can conclude that when N is large, any arbitrary variables in z can be freely set to "*absence*" or "*present*" without affecting other variables. That is, in scenarios with a large number of nodes N, it is always feasible to hold

- both $X_{i,j} = 0$ and $A_{a,b} = A_{a,b}$, as well as both $X_{i,j} = x_{i,j}$ and $A_{a,b} = 0$;
- both $A_{k,n} = 0$ and $A_{a,b} = A_{a,b}$, as well as both $A_{k,n} = A_{k,n}$ and $A_{a,b} = 0$;
- both $X_{i,j} = 0$ and $P_{c,e} = p_{c,e}$, as well as both $X_{i,j} = x_{i,j}$ and $P_{c,e} = 0$;
- both $A_{a,b} = 0$ and $P_{c,e} = p_{c,e}$, as well as both $A_{a,b} = A_{a,b}$ and $P_{c,e} = 0$;
- both $P_{c,e} = 0$ and $P_{g,h} = p_{g,h}$, as well as $P_{c,e} = p_{c,e}$ and $P_{g,h} = 0$,

without affecting other variables in z, where $X_{i,j}, A_{a,b}, A_{k,n}, P_{c,e}, P_{g,h}$ refers to the variables in z. Hence we have proved Lemma B.1. □

Next, we prove by contradiction that for all possible variable pairs (ν_i, ν_j) among the unique variables in z, we have (ν_i, ν_j) contribute equally to z at (x_i, x_j) with respect to $(0, 0)$, where $\nu_i = x_i, \nu_j = x_j$ means the "*presence*" of the variables.

Assume there exists two variables (ν_i, ν_j) in $V(z)$ that are not equally contribute to z at (x_i, x_j) with respect to $(0, 0)$. Then by Definition 1, we should have one assignment of other variables in z, such that

$$z_{\nu_i = x_i, \nu_j = 0}(V(z) \backslash \{\nu_i, \nu_j\}) \neq z_{\nu_i = 0, \nu_j = x_j}(V(z) \backslash \{\nu_i, \nu_j\}).$$

However, since

$$z_{\nu_i = x_i, \nu_j = 0}(V(z) \backslash \{\nu_i, \nu_j\}) = z_{\nu_i = 0, \nu_j = x_j}(V(z) \backslash \{\nu_i, \nu_j\}) = 0,$$

we have reached a contradiction. Hence we have proved Lemma 4. □

C Proof of Theorem 5

Proof. If the total number of unique variables in a scalar product equals to the total number of occurences of all the unique variables, i.e., if $|V(z)| = \sum_{\rho \text{ in } z} O(\rho, z)$, we will have $I_\nu(z) = \frac{z}{|V(z)|} = \frac{O(\nu, z) \cdot z}{\sum_{\rho \text{ in } z} O(\rho, z)}$. This is because when $|V(z)| = \sum_{\rho \text{ in } z} O(\rho, z)$, all the occurrences of variables are unique variables, and we have $O(\nu, z) = 1$. Consider $I_\nu(f_{m,n}(\cdot))$ as the sum of two components, which are the contribution $I_\nu(f_{m,n}^{V=O}(\cdot))$ of ν to the scalar products where $|V(z)| = \sum_{\rho \text{ in } z} O(\rho, z)$ holds, and the contribution $I_\nu(f_{m,n}^{V\neq O}(\cdot))$ of ν to the scalar products where $|V(z)| = \sum_{\rho \text{ in } z} O(\rho, z)$ does not hold. That is,

$$I_\nu(f_{m,n}(\cdot)) = I_\nu(f_{m,n}^{V=O}(\cdot)) + I_\nu(f_{m,n}^{V\neq O}(\cdot)).$$

We are not able to have multiple occurrences of X or P in a scalar product, but only able to have multiple occurrences of A. Considering the scalar products are bounded by a value of c, we have

$$\begin{aligned}\frac{I_\nu(f_{m,n}^{V\neq O}(\cdot))}{I_\nu(f_{m,n}(\cdot))} &\leq \frac{c \cdot \left[\binom{L}{2} N^{L-1} + \cdots + \binom{L}{L} N\right]}{c \cdot N^{L+1}} \\ &= \frac{L! N^{L-1}}{2!(L-2)! N^{L+1}} + \cdots + \frac{L! N}{(L)! 0! N^{L+1}} \\ &= \frac{L(L-1)}{2! N^2} + \cdots + \frac{1}{N^L} \\ &\leq \frac{L^2}{N^2} + \cdots + \frac{L^2}{N^2},\end{aligned}$$

Since $N \gg L$, we have

$$\lim_{N\to\infty} \frac{I_\nu(f_{m,n}^{V\neq O}(\cdot))}{I_\nu(f_{m,n}(\cdot))} = 0.$$

Therefore, when N is large, $I_\nu(f_{m,n}(\cdot)) = I_\nu(f_{m,n}^{V=O}(\cdot))$. Hence, by Equation (10), we have proved that when N is large, $I_\nu(f_{m,n}(\cdot)) = \sum_{z \text{ in } f_{m,n}(\cdot) \text{ that contain } \nu} \frac{O(\nu,z)}{\sum_{\rho \text{ in } z} O(\rho,z)} \cdot z.$ □

D Case study: Explaining GraphSAGE (SAmple and aggreGatE)

GraphSAGE adopts concatenation at the COMBINE step, hence the hidden state of a GraphSAGE's l-th layer is

$$H^{(l)} = \text{ReLU}\left(AH^{(l-1)}W^{(l)\phi} + H^{(l-1)}W^{(l)\psi} + B^{(l)}\right), \tag{D.1}$$

where $W^{(l)\phi}$ and $W^{(l)\psi}$ represents the trainable parameters for concatenating the node information and its neighborhood information. Suppose a GraphSAGE network $f(A, X)$ has three message-passing layers and a 2-layer MLP as the classifier, then its expansion form without the activation functions ReLU($\cdot$) will be

$$\begin{aligned} f(A, X)_{\not{P}} = {} & XW^{(1)\psi}W^{(2)\psi}W^{(3)\psi}W^{(c_1)}W^{(c_2)} + A^{(1)}XW^{(1)\phi}W^{(2)\psi}W^{(3)\psi}W^{(c_1)}W^{(c_2)} \\ & + A^{(2)}XW^{(1)\psi}W^{(2)\phi}W^{(3)\psi}W^{(c_1)}W^{(c_2)} + A^{(3)}XW^{(1)\psi}W^{(2)\psi}W^{(3)\phi}W^{(c_1)}W^{(c_2)} \\ & + A^{(2)}A^{(1)}XW^{(1)\phi}W^{(2)\phi}W^{(3)\psi}W^{(c_1)}W^{(c_2)} \\ & + A^{(3)}A^{(1)}XW^{(1)\phi}W^{(2)\psi}W^{(3)\phi}W^{(c_1)}W^{(c_2)} \\ & + A^{(3)}A^{(2)}XW^{(1)\psi}W^{(2)\phi}W^{(3)\phi}W^{(c_1)}W^{(c_2)} \\ & + A^{(3)}A^{(2)}A^{(1)}XW^{(1)\phi}W^{(2)\phi}W^{(3)\phi}W^{(c_1)}W^{(c_2)} \\ & + A^{(2)}B^{(1)}W^{(2)\phi}W^{(3)\psi}W^{(c_1)}W^{(c_2)} + A^{(3)}B^{(1)}W^{(2)\psi}W^{(3)\phi}W^{(c_1)}W^{(c_2)} \\ & + A^{(3)}A^{(2)}B^{(1)}W^{(2)\phi}W^{(3)\phi}W^{(c_1)}W^{(c_2)} + B^{(1)}W^{(2)\psi}W^{(3)\psi}W^{(c_1)}W^{(c_2)} \\ & + A^{(3)}B^{(2)}W^{(3)\phi}W^{(c_1)}W^{(c_2)} + B^{(2)}W^{(3)\psi}W^{(c_1)}W^{(c_2)} \\ & + B^{(3)}W^{(c_1)}W^{(c_2)} + B^{(c_1)}W^{(c_2)} + B^{(c_2)}. \end{aligned} \tag{D.2}$$

Then all the other steps will be identical to the case study of GCN. The code of explaining GraphSAGE on the graph classification task is in the package of Supplementary Material.

E Case Study: Explaining Graph Isomorphism Network (GIN)

GIN adopts weighted sum at the COMBINE step, hence the hidden state of a GIN's l-th layer is:

$$H^{(l)} = \Phi^{(l)}\left(\hat{A}H^{(l-1)} + \epsilon^{(l)}H^{(l-1)}\right), \tag{E.3}$$

where $\hat{A} = A + I$ refers to the adjacency matrix with the self-loops, $\epsilon^{(l)}$ is a trainable scalar parameter. If $\Phi^{(l)}(\cdot)$ is a 2-layer MLP, expanding $\Phi^{(l)}$, we have

$$H^{(l)} = \mathrm{ReLU}\left(\mathrm{ReLU}\left(\hat{A}H^{(l-1)}W^{\Phi_1^{(l)}} + \epsilon^{(l)}H^{(l-1)}W^{\Phi_1^{(l)}} + B^{\Phi_1^{(l)}}\right)W^{\Phi_2^{(l)}} + B^{\Phi_2^{(l)}}\right). \tag{E.4}$$

Suppose a GIN $f(\hat{A}, X)$ has three message-passing layers and a 2-layer MLP as the classifier, then its expansion form without the activation functions ReLU$(\cdot)$ will be

$$\begin{aligned}
f(\hat{A}, X)_{\not{P}} = {} & X\epsilon^{(3)}\epsilon^{(2)}\epsilon^{(1)}W^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(1)}X\epsilon^{(3)}\epsilon^{(2)}W^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(2)}X\epsilon^{(3)}\epsilon^{(1)}W^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(3)}X\epsilon^{(2)}\epsilon^{(1)}W^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(2)}\hat{A}^{(1)}X\epsilon^{(3)}W^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(3)}\hat{A}^{(1)}X\epsilon^{(2)}W^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(3)}\hat{A}^{(2)}X\epsilon^{(1)}W^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(3)}\hat{A}^{(2)}\hat{A}^{(1)}XW^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \epsilon^{(3)}\epsilon^{(2)}B^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(2)}\epsilon^{(3)}B^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(3)}\epsilon^{(2)}B^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(3)}\hat{A}^{(2)}B^{\Phi_1^{(1)}}W^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \epsilon^{(3)}\epsilon^{(2)}B^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(2)}\epsilon^{(3)}B^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(3)}\epsilon^{(2)}B^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(3)}\hat{A}^{(2)}B^{\Phi_2^{(1)}}W^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(3)}B^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \epsilon^{(3)}B^{\Phi_1^{(2)}}W^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + \hat{A}^{(3)}B^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} + \epsilon^{(3)}B^{\Phi_2^{(2)}}W^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} \\
& + B^{\Phi_1^{(3)}}W^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} + B^{\Phi_2^{(3)}}W^{(c_1)}W^{(c_2)} + B^{(c_1)}W^{(c_2)} + B^{(c_2)}.
\end{aligned} \tag{E.5}$$

Then similar to the case study on GCN and GraphSAGE, the activation patterns are multiplied to each of the scalar products. Although Equation (E.5) may appear complex, we can observe a pattern that when $\hat{A}^{(l)}$ is present in a product term, the corresponding $\epsilon^{(l)}$ is not. This observation allows us to simplify the expression by using for loops to cover all the product terms. The code of explaining GIN on the graph classification task is in the package of Supplementary Material.

F Handling Batch Normalization Layer

In certain cases, Batch Normalization (BN) may be applied between the message-passing layers. In this section, we will elaborate on how BN layer is handled to provide explantions with *GOAt*. The formula of BN is

$$y = \frac{x - \mu}{\sqrt{\delta + \varepsilon}} \cdot W + B, \tag{F.6}$$

where μ is the running mean, δ is the running variance, ε is a prefixed small value, W, B are learnable parameters. During the evaluation mode of a pretrained GNN, μ, δ, ε, W and B are fixed. As a result, we can treat the Batch Normalization (BN) layer as a linear mapping $y = xW^{(\mathrm{BN})} + B^{(\mathrm{BN})}$ while obtaining GNN explanations with *GOAt*, where

$$W^{(\mathrm{BN})} = \frac{W}{\sqrt{\delta + \varepsilon}},\ B^{(\mathrm{BN})} = \frac{-\mu \cdot W}{\sqrt{\delta + \varepsilon}} + B. \tag{F.7}$$

G Fidelity Results of Explaining GraphSAGE and GIN

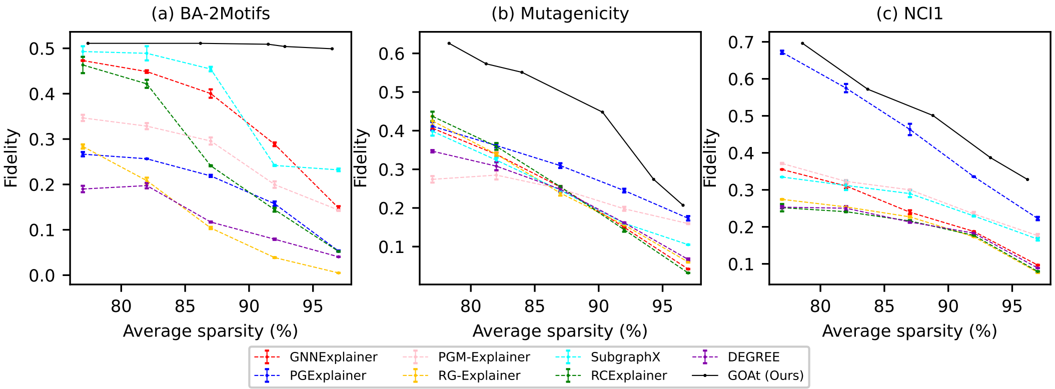

Figure G.1: Fidelity performance averaged across 10 runs on the pretrained GraphSAGE for the datasets at different levels of average sparsity.

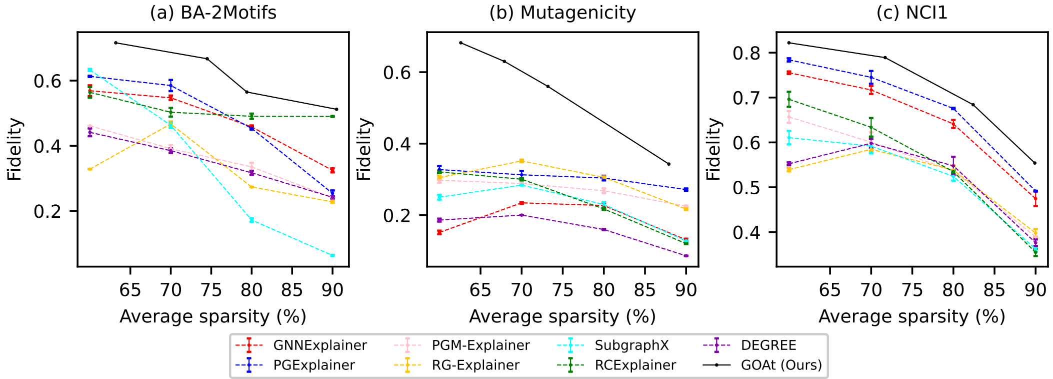

Figure G.2: Fidelity performance averaged across 10 runs on the pretrained GIN for the datasets at different levels of average sparsity.

H Statistics and Implementation Details

The GNNs are trained using the following data splits: 80% for the training set, 10% for the validation set, and 10% for the testing set. All experiments are conducted on an Intel® Core™ i7-10700 Processor and NVIDIA GeForce RTX 3090 Graphics Card. The GNN architectures consist of 3 message-passing layers and a 2-layer classifier. The hidden dimension is set to 32 for BA-2Motifs, BA-Shapes, BA-Community, Tree-grid, and 64 for Mutagenicity and NCI1. The code is available in the Supplementary Material, provided alongside this Appendix file.

Table H.1: Statistics of the datasets used and the classification accuracy of the trained GNNs.

		BA-Shapes	BA-Community	Tree-Grid	BA-2Motifs	Mutagenicity	NCI1
# Graphs		1	1	1	1,000	4,337	4,110
# Nodes (avg)		700	1,400	1,231	25	30.32	29.87
# Edges (avg)		4,110	8,920	3,410	25.48	30.77	32.30
# Classes		4	8	2	2	2	2
Test ACC	GCN	0.97	0.91	0.97	1.00	0.82	0.81
	GraphSAGE	-	-	-	1.00	0.80	0.80
	GIN	-	-	-	1.00	0.89	0.83

I Visualization of Explanation Embeddings on Mutagenicity and NCI1

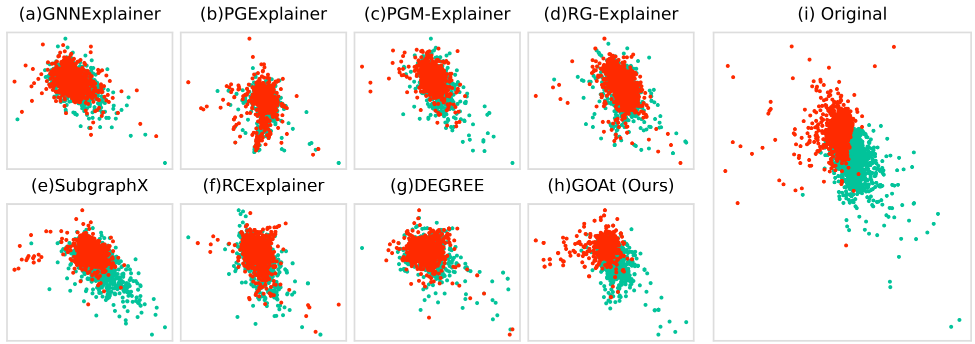

Figure I.3: Visualization of explanation embeddings on the Mutagenicity dataset. Subfigure (i) refers to the visualization of the original embeddings by directly feeding the original data into the GNN without any modifications or explanations applied.

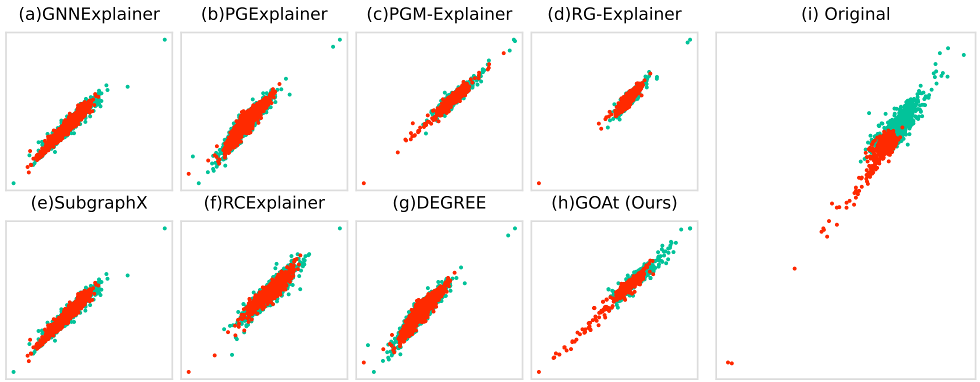

Figure I.4: Visualization of explanation embeddings on the NCI1 dataset. Subfigure (i) refers to the visualization of the original embeddings by directly feeding the original data into the GNN without any modifications or explanations applied.

Figure I.3 and Figure I.4 presented the visualization of explanation embeddings on the Mutagenicity and NCI1 datasets respectively. To create smaller plots, we have disabled the axes in Figure I.3 and Figure I.4. In Figure I.5, we have enabled the axes for specific subplots to showcase the dispersion of explanation embeddings from *GOAt* compared to SubgraphX and the original embeddings. Specifically, in the case of Mutagenicity, although the explanations generated by SubgraphX exhibit only some overlap, the scatters for different classes appear quite close. On the other hand, *GOAt* produces more discriminative explanations. For the NCI1 dataset, while the majority of explanations generated by SubgraphX overlap, the explanations from *GOAt* exhibit greater dispersion in the scatter plot. Furthermore, compared to the original embeddings, the explanations generated by *GOAt* for

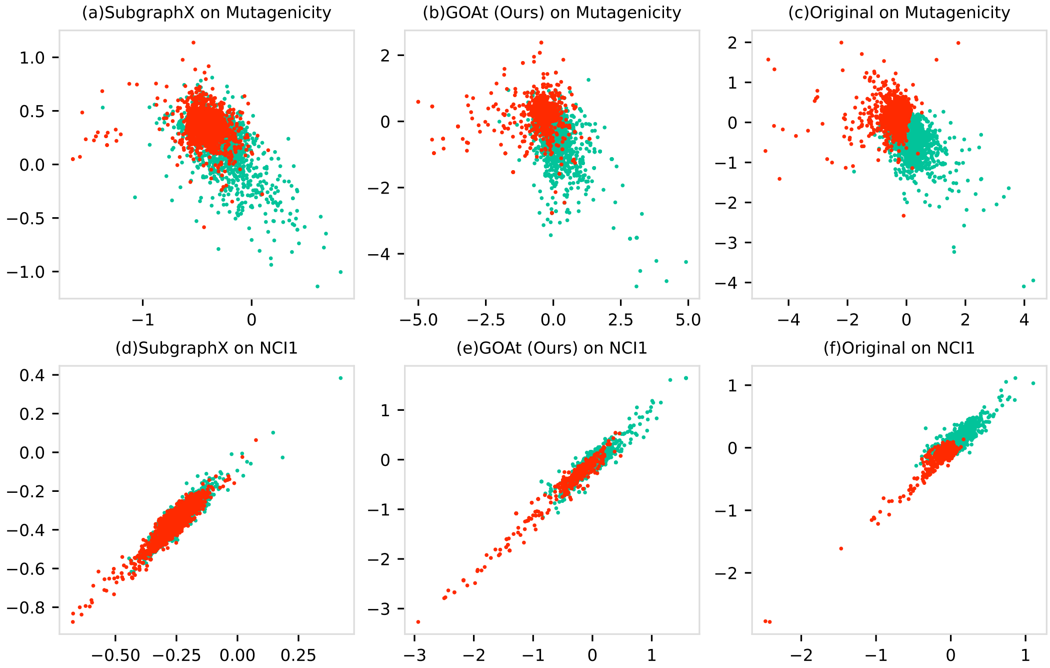

Figure I.5: Visualization of explanation embeddings on the Mutagenicity and NCI1 datasets with axes turned on.

NCI1 demonstrate higher confidence towards specific classes, as evident from the bottom-left area in Figure I.5(e).

J Explanations of Node Classification

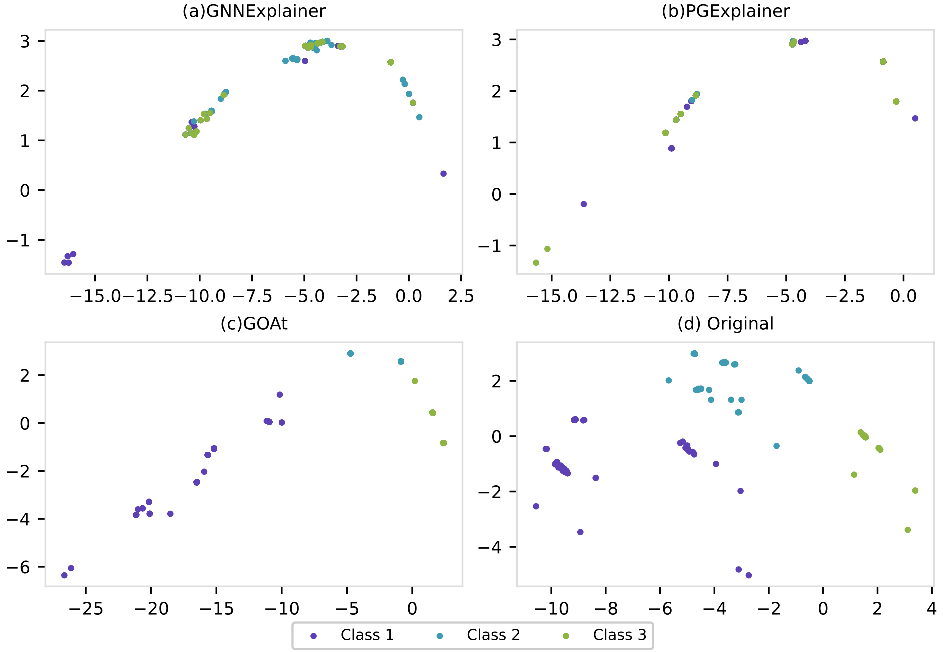

Figure J.6: Visualization of explanation embeddings on the BA-Shapes dataset. Subfigure (d) refers to the visualization of the original embeddings by directly feeding the original data into the GNN without any modifications or explanations applied.

We utilize scatter plots to visually depict the explanation embeddings produced by GNNExplainer, PGExplainer, and GOAt, and compare them with the node embeddings in the original graphs. In the generated figures, we set the value of $topk$ to be 6, 7, and 14 for the BA-Shapes, BA-Community, and Tree-Grid datasets, respectively. In the case of BA-Shapes and BA-Community, we only plot the nodes within the house-shape motif, as the other nodes are located far away and may not be easily discernible in terms of explanation performance.

As presented in Figure J.8, the majority of the explanations on the Tree-Grid dataset generated by GNNExplainer are closely clustered together, and *GOAt* has fewer overlapped data points than PGExplainer. As illustrated in Figure J.6 and Figure J.7, the explanations generated by GNNExplainer and PGExplainer fail to exhibit class discrimination on BA-Shapes and BA-Communicty datasets, as all the data points are clustered together without any distinct separation. In contrast, our method, *GOAt*, generates explanations that clearly and effectively distinguish between classes, with fewer overlapping points and substantial separation distances, highlighting the strong discriminability of our approach on the node classification task.

Discussion on the AUC/Accuracy metrics. Many existing GNN explanation approaches are evaluated using metrics such as AUC or Accuracy. These metrics compare the explanations generated by the explainers with "ground-truth" explanations that are predetermined by humans. Ground-truth explanations refer to the underlying evidence that leads to the correct label, rather than the prediction label itself. However, as highlighted by [1] there can often be a mismatch between the ground truth and the GNN. To avoid any potential misunderstandings, we have chosen to directly present scatter plots of the explanations generated by different explainers.

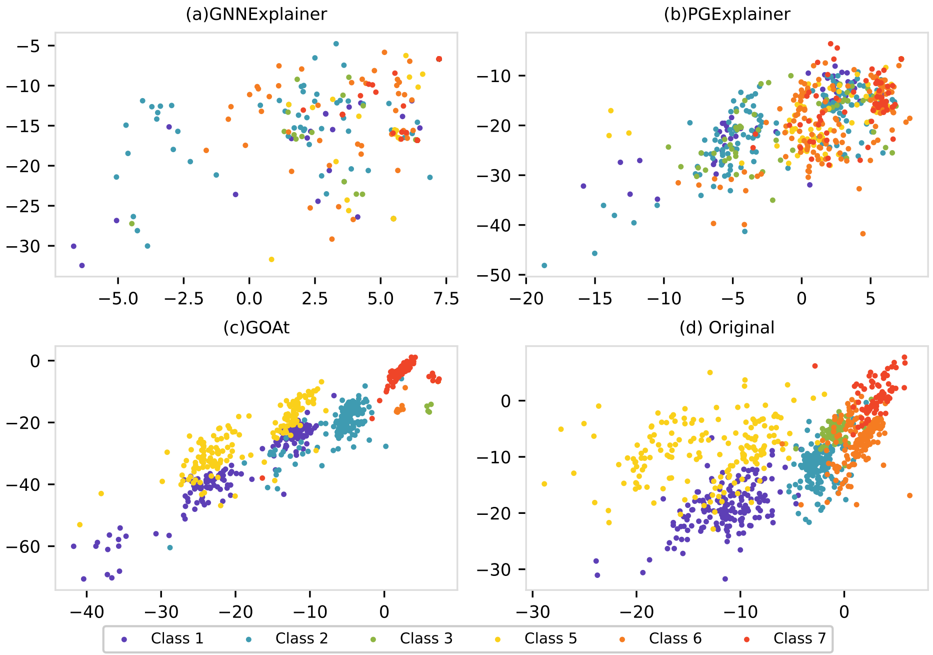

Figure J.7: Visualization of explanation embeddings on the BA-Community dataset. Subfigure (d) refers to the visualization of the original embeddings by directly feeding the original data into the GNN without any modifications or explanations applied.

K Broader Impacts and Limitations

Our technique aims to contribute to the community's understanding of the decision-making process in GNNs and enhance the reliability of these models. We hope that our approach will be valuable in advancing the field and fostering greater trust and transparency in GNNs. The core concept of our proposed GNN explaining approach, *GOAt*, which is based on "Equal Contribution in the scalar product," can potentially be extended to explain other neural networks, including CNNs. However,

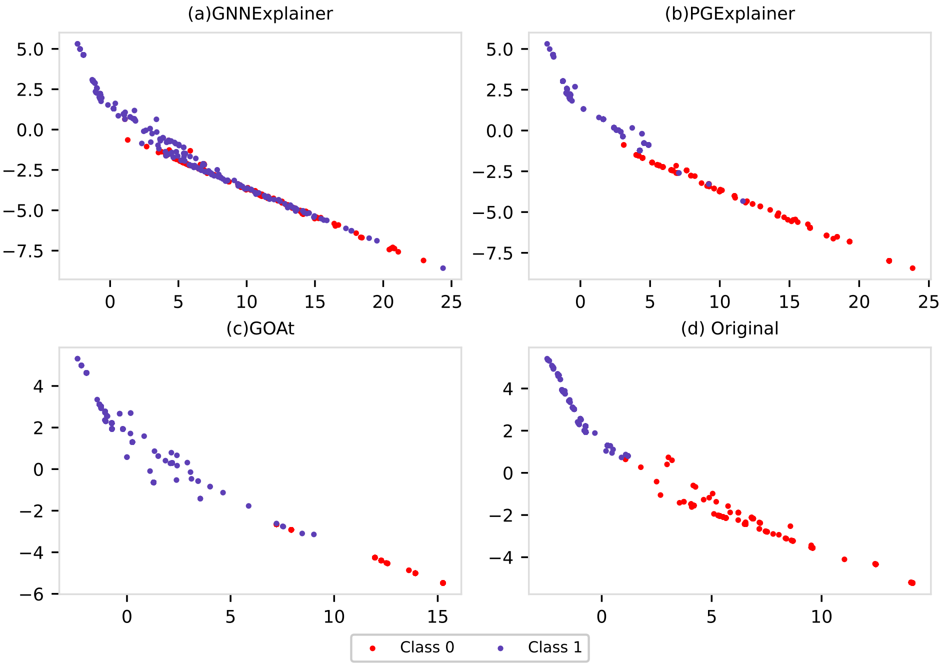

Figure J.8: Visualization of explanation embeddings on the Tree-Grid dataset. Subfigure (d) refers to the visualization of the original embeddings by directly feeding the original data into the GNN without any modifications or explanations applied.

it is important to note that our technique currently requires expert knowledge to design specific explaining flows for different neural network architectures. While our method works well for shallow networks like GNNs, it may become more challenging for deeper networks such as Transformers or ResNets, where the explaining flows can become complex. In such cases, it may be necessary to group or prune scalar products that contribute minimally to the outputs. These are potential areas for future research and investigation.

References

[1] Lukas Faber, Amin K. Moghaddam, and Roger Wattenhofer. When comparing to ground truth is wrong: On evaluating gnn explanation methods. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 332–341, 2021.