Supplementary Material on Methodology

Symbol Summary

Table 6 provides a summary of all symbols used in this paper, along with their descriptions.

$\begin{array}{cccccccccccccccccccccccccccccccccccc$	Symbol	Description
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$		
$\begin{array}{lll} A & {\rm Adjacency\ matrix} \\ n & {\rm Number\ of\ nodes} \\ m & {\rm Number\ of\ edges} \\ m & {\rm Number\ of\ edges} \\ m & {\rm Number\ of\ edges} \\ M & {\rm Target\ GNN\ model} \\ Y & {\rm Graph\ node\ classification\ label} \\ E & {\rm Environment\ variables} \\ G_c & {\rm Explanation\ subgraph} \\ G_s & {\rm Graph\ in\ testing\ dataset} \\ X_{str} & {\rm Structure\ based\ efastures\ for\ nodes} \\ H_{str} & {\rm Structure\ based\ embedding\ for\ nodes} \\ H_{str} & {\rm Structure\ based\ embedding\ for\ praphs} \\ K & {\rm Number\ of\ potential\ environments} \\ E_{nv} & {\rm Environmental\ embedding\ ser} \\ E_{tr} & {\rm Structure\ based\ emvironment\ label\ set} \\ E_{tr} & {\rm Thc\ k\ th\ structure\ based\ environment\ label} \\ E_{tr} & {\rm Thc\ k\ th\ structure\ based\ environment\ label} \\ E_{tr} & {\rm Thc\ k\ th\ fature\ based\ environment\ label} \\ E_{tr} & {\rm Thc\ k\ th\ fature\ based\ environment\ label} \\ E_{tr} & {\rm Thc\ k\ th\ fature\ based\ environment\ label} \\ E_{tr} & {\rm Thc\ k\ th\ fature\ based\ environment\ label} \\ E_{tr} & {\rm Thc\ k\ th\ fature\ based\ environment\ label} \\ E_{tr} & {\rm Thc\ k\ th\ fature\ based\ environment\ label} \\ E_{tr} & {\rm Thc\ k\ th\ fature\ based\ environment\ label} \\ V_c & {\rm Nodes\ causally\ related\ to\ th\ classification\ label} \\ N_{tr} & {\rm Causall\ p\ related\ to\ th\ classification\ label} \\ N_{tr} & {\rm Fatur\ based\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\ G} \\ e_i & {\rm Environmental\ embedding\ for\ raph\$		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	\mathcal{X}	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	A	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
$\begin{array}{lll} G_{\rm ext} & {\rm Graph \ in \ testing \ dataset} \\ X_{str} & {\rm Structure-based \ entures \ for \ nodes} \\ H_{str} & {\rm Structure-based \ embeddings \ for \ nodes} \\ H_G & {\rm Structure-based \ embeddings \ for \ graphs} \\ K & {\rm Number \ of \ potential \ environments} \\ env & {\rm Environmental \ embedding \ set} \\ E_{str} & {\rm Structure-based \ emvironment \ label \ set} \\ E_k & {\rm The \ k-th \ feature-based \ environment \ label \ set} \\ E_k & {\rm The \ k-th \ feature-based \ environment \ label} \\ E_{feat} & {\rm Feature-based \ environment \ label} \\ V_c & {\rm Nodes \ causally \ related \ to \ the \ environment} \\ V_s & {\rm Nodes \ causally \ related \ to \ the \ environment} \\ H & {\rm Feature-based \ embedding \ for \ node \ i} \\ V_s & {\rm Nodes \ causally \ related \ to \ the \ environment} \\ h_i & {\rm Feature-based \ embedding \ for \ node \ i} \\ h_i & {\rm Feature-based \ embedding \ for \ node \ i} \\ e_i & {\rm Environmental \ embedding \ for \ node \ i} \\ e_i & {\rm Environmental \ embedding \ for \ node \ i} \\ e_i & {\rm Environmental \ embedding \ for \ node \ i} \\ e_i & {\rm Environmental \ embedding \ for \ node \ i} \\ e_i & {\rm Environmental \ embedding \ for \ node \ i} \\ e_i & {\rm Environmental \ embedding \ for \ node \ i} \\ e_i & {\rm Environmental \ embedding \ for \ node \ i} \\ e_i & {\rm Environmental \ embedding \ for \ graph \ G} \\ \mu_i & {\rm Mean \ of \ the \ node-invariant \ representation \ distribution \ of \ node \ i} \\ e_i & {\rm Environmental \ embedding \ for \ parph \ G} \\ e_i & {\rm Environmental \ embedding \ for \ parph \ G} \\ E_i & {\rm Environmental \ embedding \ for \ node \ i} \\ e_i & {\rm Environmental \ embedding \ for \ parph \ G} \\ E_i & {\rm Environmental \ embedding \ for \ parph \ G} \\ E_i & {\rm Environmental \ embedding \ for \ parph \ G} \\ E_i & {\rm Environmental \ embedding \ for \ parph \ G} \\ E_i & {\rm Environmental \ embedding \ for \ parph \ G} \\ E_i & {\rm Environmental \ embedding \ for \ parph \ G} \\ E_i & {\rm Environmental \ embedding \ for \ parph \ G} \\ E_i & {\rm Environmental \ embedding $		
$\begin{array}{lll} X_{str} & \text{Structure-based embeddings for nodes} \\ H_G & \text{Structure-based embeddings for nodes} \\ K & \text{Number of potential environments} \\ env & \text{Environmental embedding set} \\ E_{str} & \text{Structure-based environment label set} \\ E_{feat} & \text{Environmental embedding set} \\ E_{feat} & \text{The k-th structure-based environment label} \\ E_{feat} & \text{Feature-based environment label} \\ V_c & \text{Nodes causally related to the classification label} \\ V_s & \text{Nodes causally related to the environment} \\ H & \text{Feature-based embedding for node i} \\ h_i & \text{Feature-based embedding for node i} \\ h_G & \text{Feature-based embedding for graph G} \\ e_i & \text{Environmental embedding for node i} \\ h_G & \text{Feature-based embedding for node i} \\ e_G & \text{Environmental embedding for node i} \\ e_G & \text{Log-variance of the node-invariant representation distribution of node i} \\ e_G & \text{Node-invariant embedding for graph G} \\ e_G & \text{Random noise} \\ e_G & \text{Random noise} \\ e_G & \text{Graph-invariant embedding for graph G} \\ e_G & \text{Random noise} \\ \text{Since}(i_1) & \text{Node-invariant embedding of graph G} \\ e_G & \text{Random noise} \\ \text{Since}(i_2) & \text{Since}(i_3) & \text{Since}(i_4) \\ e_{g_2}(\mathbf{z}_G G, env) & \text{Distribution modeled by the NodeVAE encoder} \\ p_{g_2}(\mathbf{z}_G G, env) & \text{Distribution modeled by the NodeVAE encoder} \\ p_{g_3}(e_{ij} \mathbf{z}_G, \mathbf{z}_i, env) & \text{Distribution modeled by the NodeVAE encoder} \\ p_{g_3}(e_{ij} \mathbf{z}_G, \mathbf{z}_i, env) & \text{Distribution modeled by the NodeVAE encoder} \\ p_{g_3}(e_{ij} \mathbf{z}_G, \mathbf{z}_i, env) & \text{Distribution modeled by the NodeVAE encoder} \\ p_{g_3}(e_{ij} \mathbf{z}_G, \mathbf{z}_i, env) & \text{Distribution modeled by the NodeVAE encoder} \\ p_{g_3}(e_{ij} \mathbf{z}_G, \mathbf{z}_i, env) & Distribution modeled by the NodeVA$		
$\begin{array}{lll} H_{atr} & \text{Structure-based embeddings for nodes} \\ H_G & \text{Structure-based embeddings for graphs} \\ K & \text{Number of potential environments} \\ env & \text{Environmental embedding set} \\ E_{str} & \text{Structure-based environment label set} \\ E_{st} & \text{The k-th structure-based environment label} \\ E_{feat} & \text{Feature-based environment label set} \\ E_k^f & \text{The k-th feature-based environment label} \\ V_c & \text{Nodes causally related to the classification label} \\ V_s & \text{Nodes causally related to the environment} \\ H & \text{Feature-based embedding for node i} \\ H_G & \text{Feature-based embedding for node i} \\ e_i & \text{Environmental embedding for node i} \\ e_G & \text{Environmental embedding for node i} \\ e_G & \text{Environmental embedding for graph G} \\ e_i & \text{Environmental embedding for graph G} \\ e_G & \text{Environmental embedding for graph G} \\ \text{Mean of the graph-invariant representation distribution of node i} \\ \log(\sigma_0^2) & \text{Log-variance of the node-invariant representation distribution of node i} \\ \log(\sigma_0^2) & \text{Log-variance of the node-invariant representation distribution of fonde i} \\ \text{Z}_{node} & \text{Node-invariant representation of node i} \\ \text{Z}_{node} & Node-invariant representation no$	X	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	H_{str}	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	K	Number of potential environments
$\begin{array}{c c} E_k^k \\ E_{feat} \\ E_{fe$	env	Environmental embedding set
$\begin{array}{c c} E_{feat}^* & \text{Feature-based environment label set} \\ E_k^* & \text{The k-th feature-based environment label} \\ Y_c & \text{Nodes causally related to the classification label} \\ Y_s & \text{Nodes causally related to the environment} \\ \mathbf{H} & \text{Feature-based onde embedding} \\ \mathbf{h}_i & \text{Feature-based embedding for node i} \\ \mathbf{h}_G & \text{Feature-based embedding for node i} \\ \mathbf{e}_G & \text{Environmental embedding for graph G} \\ \mathbf{e}_i & \text{Environmental embedding for node i} \\ \mathbf{e}_G & \text{Environmental embedding for graph G} \\ \boldsymbol{\mu}_i & \text{Mean of the node-invariant representation distribution of node i} \\ \boldsymbol{\mu}_G & \text{Mean of the node-invariant representation distribution of node i} \\ \mathbf{log}(\boldsymbol{\sigma}_G^2) & \text{Log-variance of the node-invariant representation distribution of node i} \\ \mathbf{log}(\boldsymbol{\sigma}_G^2) & \text{Log-variance of the graph-invariant representation distribution of graph G} \\ \boldsymbol{\lambda}_G & \text{Node-invariant representation of node i} \\ \mathbf{\lambda}_{G}(\boldsymbol{\beta}_i) & \text{Node-invariant mebedding of graph G} \\ \boldsymbol{\lambda}_{G}(\boldsymbol{\beta}_i) & \text{Node-invariant embedding of graph G} \\ \boldsymbol{\lambda}_{G}(\boldsymbol{\beta}_i) & \text{Is divergence} \\ \mathbf{\lambda}_{G}(\boldsymbol{\beta}_i) & \text{Is divergence} \\ \mathbf{\lambda}_{G}(\boldsymbol{\beta}_i, \boldsymbol{\epsilon}_i, \boldsymbol{\epsilon}_i,$		
$\begin{array}{c} E_k^f \\ \mathcal{V}_c \\ \mathcal{V}_c \\ \mathcal{V}_s \\ \mathcal{V}_s$		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	E_k^j	
$\begin{array}{lll} \mathbf{H} & \text{Feature-based node embeddings} \\ \mathbf{h}_i & \text{Feature-based embedding for node } i \\ \mathbf{h}_G & \text{Feature-based embedding for graph } G \\ \mathbf{e}_i & \text{Environmental embedding for graph } G \\ \mathbf{e}_G & \text{Environmental embedding for graph } G \\ \boldsymbol{\mu}_i & \text{Mean of the node-invariant representation distribution of node } i \\ \boldsymbol{\mu}_G & \text{Mean of the graph-invariant representation distribution of node } i \\ \log(\sigma_i^2) & \text{Log-variance of the node-invariant representation distribution of node } i \\ \log(\sigma_G^2) & \text{Log-variance of the graph-invariant representation distribution of graph } G \\ \mathbf{z}_i & \text{Node-invariant representation of node } i \\ \mathbf{z}_{node} & \text{Node-invariant representation of node } i \\ \mathbf{z}_{node} & \text{Node-invariant mebeddings} \\ \mathbf{z}_G & \text{Graph-invariant embedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant embedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & \text{Node-invariant mebedding of graph } G \\ \mathbf{z}_i & Node-Node-invariant mebedding of graph invariant methedding of graph invarian$	V_c	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
$\begin{array}{lll} \mathbf{e}_{G} & \operatorname{Environmental embedding for graph } G \\ \boldsymbol{\mu}_{i} & \operatorname{Mean of the node-invariant representation distribution of node } i \\ \boldsymbol{\mu}_{G} & \operatorname{Mean of the graph-invariant representation distribution of } G \\ \log(\sigma_{c}^{2}) & \operatorname{Log-variance of the node-invariant representation } \\ \operatorname{distribution of node } i \\ \log(\sigma_{G}^{2}) & \operatorname{Log-variance of the node-invariant representation } \\ \operatorname{distribution of node } i \\ \operatorname{Log-variance of the graph-invariant representation } \\ \operatorname{distribution of node } i \\ \operatorname{Log-variance of the graph-invariant representation } \\ \operatorname{distribution of graph } G \\ \mathbf{z}_{i} & \operatorname{Node-invariant embeddings} \\ \mathbf{z}_{G} & \operatorname{Graph-invariant embeddings} \\ \mathbf{z}_{G} & \operatorname{Sinder} \mathbf{z}_{G} \\ \mathbf{z}_{i} & \operatorname{Invariant embeddings} \\ \mathbf{z}_{G} & \operatorname{Sinder} \mathbf{z}_{G} \\ \mathbf{z}_{i} & \operatorname{Invariant embeddings} \\ \mathbf{z}_{G} & \operatorname{Ginder} \mathbf{z}_{G} \\ \mathbf{z}_{i} & \operatorname{Invariant embeddings} \\ \mathbf{z}_{G} & \operatorname{Ginder} \mathbf{z}_{G} \\ \mathbf{z}_{i} & \operatorname{Invariant embeddings} \\ \mathbf{z}_{G} & \operatorname{Ginder} \mathbf{z}_{G} \\ \mathbf{z}_{i} & \operatorname{Invariant embeddings} \\ \mathbf{z}_{G} & \operatorname{Invariant embeddings} $		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
$\log(\sigma_c^2) \qquad \text{Log-variance of the node-invariant representation distribution of node } i \\ \log(\sigma_G^2) \qquad \text{Log-variance of the graph-invariant representation } \\ \log(\sigma_G^2) \qquad \text{Log-variance of the graph-invariant representation } \\ \log(\sigma_G^2) \qquad \text{Log-variance of the graph-invariant representation } \\ \log(\sigma_G^2) \qquad \text{Log-variance of the graph-invariant representation } \\ \log(\sigma_G^2) \qquad \text{Node-invariant representation of node } i \\ \log(\sigma_G^2) \qquad \text{Node-invariant embeddings} \\ \log(\sigma_G^2) \qquad \text{Node-invariant embedding of graph } G \\ \log(\sigma_G^2) \qquad \text{Exp. of the node-invariant representation of node } i \\ \log(\sigma_G^2) \qquad \text{Node-invariant embeddings} \\ \log(\sigma_G^2) \qquad \text{Exp. of the node-invariant representation of node } i \\ \log(\sigma_G^2) \qquad \text{Exp. of the node-invariant embeddings} \\ \log(\sigma_G^2) \qquad \text{Exp. of the node-invariant embeddings} \\ \log(\sigma_G^2) \qquad \text{Node-VAE pooled} \\ \log(\sigma_G^2) \qquad \text{Exp. of the node-invariant embeddings} \\ \log(\sigma_G^2) \qquad Exp. of t$	• •	
$\log(\sigma_c^2) \qquad \text{Log-variance of the node-invariant representation distribution of node } i$ $\log(\sigma_G^2) \qquad \text{Log-variance of the graph-invariant representation distribution of graph } G$ $\mathbf{z}_i \qquad \text{Node-invariant representation of node } i$ $\mathbf{z}_{\text{node}} \qquad \text{Node-invariant representation of node } i$ $\mathbf{z}_{\text{node}} \qquad \text{Node-invariant rembeddings}$ $\mathbf{z}_G \qquad \text{Graph-invariant embedding of graph } G$ $\mathbf{z}_G \qquad \text{Random noise}$ $\mathbf{K} \mathbf{L} \text{ divergence}$ $\mathbf{J} \mathbf{S}(\cdot \cdot) \qquad \mathbf{S} \text{ divergence}$ $\mathbf{J} \mathbf{S}(\cdot \cdot) \qquad \mathbf{S} \text{ divergence}$ $\mathbf{J} \mathbf{S} \text{ divergence}$ Distribution modeled by the NodeVAE encoder pistribution modeled by the NodeVAE decoder pistribution modeled by the GVAG encoder pose of the decoder pose probability distribution modeled by GVAG decoder pistribution modeled by GVAG decoder probability of graph-invariant representations prob(e_{ij}) \qquad \mathbf{L}_{\text{dist}} \text{ Subgraph reconstruction loss} \mathbf{M} \mathbf{I} \text{ loss} $\mathcal{L}_{\text{RR}} \text{ Reconstruction regularization loss}$ $\mathcal{L}_{\text{NodeVAE}} \text{ NodeVAE loss}$ $\mathcal{R}_{\text{causal}} \text{ Causal structure regularization}$ $\mathcal{R}_{\text{hinge}} \text{ Hinge regularization}$ $\mathcal{R}_{\text{hinge}} \text{ Hinge regularization}$ $\mathcal{R}_{\text{subg,node}} \text{ Subgraph node count regularization}$ $\mathcal{L}_{\text{CCON}} \text{ Contrastive loss}$ $\text{LAR} \text{ Last action rewards}$ $\text{fid}_{-} \text{ Nogative Fidelity}$ $\text{fid}_{+} \text{ Positive Fidelity}$ $\text{GEF} \text{ Unfaithfulness}$ $\rho_v \text{ Node density relative to the original graph}$ $\text{Edge density relative to the original graph}$	$oldsymbol{\mu}_G$	Mean of the graph-invariant representation distribution of G
$\log(\sigma_G^2) \qquad \text{Log-variance of the graph-invariant representation distribution of graph } G$ $\mathbf{z}_i \qquad \text{Node-invariant representation of node } i$ $\mathbf{z}_{\text{Conde}} \qquad \text{Node-invariant embeddings}$ $\mathbf{z}_G \qquad \text{Graph-invariant embeddings}$ $\mathbf{z}_G \qquad \text{Graph-invariant embedding of graph } G$ $\mathbf{z}_{\text{Random noise}} \qquad \text{Random noise}$ $D_{\text{KL}}(\cdot \parallel \cdot) \qquad \text{KL divergence}$ $JS (\cdot \parallel \cdot) \qquad \text{SK L divergence}$ $JS \text{ diversence}$ $JS \text{ diversence}$	$\log(\sigma_i^2)$	Log-variance of the node-invariant representation
distribution of graph G $\mathbf{z}_{i} \qquad \text{Node-invariant representation of node } i$ $\mathbf{z}_{\text{node}} \qquad \text{Node-invariant rembeddings}$ $\mathbf{z}_{G} \qquad \text{Graph-invariant embeddings}$ $\mathbf{z}_{G} \qquad \text{Graph-invariant embedding of graph } G$ $\mathbf{z}_{G} \qquad \text{Random noise}$ $KL \ \text{divergence}$ $IS \ \text{diverser}$ $IS \ \text{diverser}$ $IS \ \text{diverser}$ $IS \ \text{diverser}$ $IS \ di$		
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\log(\boldsymbol{\sigma}_G^2)$	
$\begin{array}{c c} \mathbf{z}_{\text{node}} & \text{Node-invariant embeddings} \\ \mathbf{z}_{G} & \text{Graph-invariant embedding of graph } G \\ \mathbf{\epsilon} & \text{Random noise} \\ D_{\text{KL}}(\cdot \parallel \cdot) & \text{KL divergence} \\ JS(\cdot \parallel \cdot) & \text{KL divergence} \\ JS \text{ divergence} & \text{JS divergence} \\ JS \text{ divergence} & \text{JS divergence} \\ p_{\theta_1}(\mathbf{h}_i \mid \mathbf{z}_i, env) & \text{Distribution modeled by the NodeVAE encoder} \\ p_{\theta_2}(\mathbf{z}_G \mid G_i, env) & \text{Distribution modeled by the NodeVAE decoder} \\ p_{\theta_2}(\mathbf{v}_i \mid \mathbf{z}_G, \mathbf{z}_i, \mathbf{z}_v) & \text{Distribution modeled by the GVAG encoder} \\ p_{\theta_3}(\mathbf{e}_{ij} \mid \mathbf{z}_G, \mathbf{z}_i, \mathbf{z}_j, env) & \text{Node existence probability distribution modeled by } \\ \text{GVAG decoder} \\ p(\mathbf{z}) & \text{Prob}(\mathbf{v}_i) & \text{Node existence probability of node } i \\ \text{Prob}(\mathbf{e}_{ij}) & \text{Edge existence probability of edge } \mathbf{e}_{ij} \\ \mathcal{L}_{\text{MI}} & \text{Subgraph reconstruction loss} MI \text{ loss} \\ \mathcal{L}_{\text{RR}} & \text{Reconstruction regularization loss} \\ \mathcal{R}_{\text{causal}} & \text{Causal structure regularization} \\ \mathcal{R}_{\text{hinge}} & \text{Hinge regularization} \\ \mathcal{R}_{\text{hinge}} & \text{Subgraph node count regularization} \\ \mathcal{L}_{\text{CON}} & \text{Contrastive loss} \\ \text{LAR} & \text{Last action rewards} \\ \text{fid} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_v & \text{Node density relative to the original graph} \\ \mathcal{L}_{\text{Edge density relative to the original graph} \\ \mathcal{L}_{Edge density relative tor$		
$ \begin{array}{c c} \mathbf{z}_{G} & \operatorname{Graph-invariant embedding of graph } G \\ \boldsymbol{\epsilon} & \operatorname{Random noise} \\ N_{\mathrm{RL}}(\cdot \parallel \cdot) & \operatorname{KL \ divergence} \\ JS(\cdot \parallel \cdot) & \operatorname{IS \ divergence} \\ J_{g_{\phi_{1}}}(\mathbf{z}_{i} \mid \mathbf{h}_{i}, env) & \operatorname{Distribution modeled by the \ NodeVAE \ encoder} \\ p_{\theta_{1}}(\mathbf{h}_{i} \mid \mathbf{z}_{i}, env) & \operatorname{Distribution modeled by the \ NodeVAE \ encoder} \\ p_{\theta_{2}}(\mathbf{v}_{i} \mid \mathbf{z}_{G}, \mathbf{env}) & \operatorname{Distribution modeled \ by the \ GVAG \ encoder} \\ p_{\theta_{3}}(\mathbf{e}_{ij} \mid \mathbf{z}_{G}, \mathbf{z}_{i}, \mathbf{env}) & \operatorname{Node \ existence \ probability \ distribution \ modeled \ by} \\ GVAG \ decoder \\ p_{\theta_{3}}(\mathbf{e}_{ij} \mid \mathbf{z}_{G}, \mathbf{z}_{i}, \mathbf{z}_{j}, env) & \operatorname{GVAG \ decoder} \\ p(\mathbf{z}) & \operatorname{Frob}(\mathbf{e}_{ij}) & \operatorname{Edge \ existence \ probability \ distribution \ modeled \ by} \\ GVAG \ decoder & \operatorname{Prob}(\mathbf{e}_{ij}) & \operatorname{Edge \ existence \ probability \ of \ node \ i} \\ Prob(\mathbf{e}_{ij}) & \operatorname{Edge \ existence \ probability \ of \ node \ i} \\ Prob(\mathbf{e}_{ij}) & \operatorname{Edge \ existence \ probability \ of \ node \ i} \\ \mathcal{L}_{RR} & \operatorname{Reconstruction \ loss} MI \ loss \\ \mathcal{R}_{\text{Causal}} & \operatorname{NodeVAE} \ loss \\ \mathcal{R}_{\text{Causal}} & \operatorname{NodeVAE} \ loss \\ \mathcal{R}_{\text{Causal}} & \operatorname{Causal \ structure \ regularization} \\ \mathcal{R}_{\text{hinge}} & \operatorname{Hinge \ regularization} \\ \mathcal{R}_{\text{subg}, \text{node}} & \operatorname{Subgraph \ node \ count \ regularization} \\ \mathcal{L}_{\text{CON}} & \operatorname{Contrastive \ loss} \\ \operatorname{LAR} & \operatorname{Last \ action \ rewards} \\ \operatorname{idd} & \operatorname{Negative \ Fidelity} \\ \operatorname{fid}_{+} & \operatorname{Positive \ Fidelity} \\ \operatorname{GEF} & \operatorname{Unfaithfulness} \\ \rho_{v} & \operatorname{Node \ density \ relative \ to \ the \ original \ graph} \\ \mathcal{E}_{\text{deg}} & \operatorname{density \ relative \ to \ the \ original \ graph} \\ \mathcal{G}_{\text{deg}} & \operatorname{dege \ density \ relative \ to \ the \ original \ graph} \\ \mathcal{G}_{\text{deged}} & \operatorname{degel}_{\text{deged}} & \operatorname{degel}_{\text{deged}} \\ \mathcal{G}_{\text{deged}} & \operatorname{degel}_{\text{deged}} & \operatorname{degel}_{de$		
$\begin{array}{lll} \epsilon & \text{Random noise} \\ D_{\text{KL}}(\cdot \parallel \cdot) & \text{KL divergence} \\ JS(\cdot \parallel \cdot) & \text{IS divergence} \\ q_{\phi_1}(\mathbf{z}_i \mathbf{h}_i, env) & \text{Distribution modeled by the NodeVAE encoder} \\ p_{\theta_1}(\mathbf{h}_i \mathbf{z}_i, env) & \text{Distribution modeled by the NodeVAE decoder} \\ p_{\theta_2}(\mathbf{z}_G G, env) & \text{Distribution modeled by the NodeVAE decoder} \\ p_{\theta_2}(\mathbf{v}_i \mathbf{z}_G, \mathbf{z}_i, env) & \text{Distribution modeled by the NodeVAE decoder} \\ p_{\theta_3}(e_{ij} \mathbf{z}_G, \mathbf{z}_i, \mathbf{z}_j, env) & \text{Node existence probability distribution modeled by } \\ \mathbf{p}(\mathbf{z}) & \text{Prob}(v_i) & \text{Edge existence probability of inde } i \\ \mathbf{p}(\mathbf{z}) & \text{Prob}(v_i) & \text{Node existence probability of edge } e_{ij} \\ \mathbf{z}_{\text{MI}} & \text{Subgraph reconstruction loss} MI \text{ loss} \\ \mathbf{z}_{\text{RR}} & \text{Reconstruction regularization loss} \\ \mathbf{z}_{\text{Causal}} & \text{NodeVAE loss} \\ \mathbf{z}_{\text{Causal}} & \text{Causal structure regularization} \\ \mathbf{z}_{\text{shipe}} & \text{node} \\ \mathbf{z}_{\text{CON}} & \text{Contrastive loss} \\ \mathbf{LAR} & \text{Last action rewards} \\ \text{fid} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_v & \text{Node density relative to the original graph} \\ \mathbf{z}_{\text{deg}} & \text{easign} \\ \mathbf{z}_{\text{def}} & \text{easign} \\ \mathbf{z}_{de$		
$\begin{array}{c c} D_{\mathrm{KL}}(\cdot \parallel \cdot) \\ JS(\cdot \parallel \cdot) \\ p_{\theta_1}(\mathbf{z}_i \parallel \mathbf{h}_i, env) \\ p_{\theta_1}(\mathbf{z}_i \parallel \mathbf{h}_i, env) \\ p_{\theta_2}(\mathbf{z}_i \parallel \mathbf{h}_i, env) \\ p_{\theta_3}(\mathbf{z}_i \parallel \mathbf{h}_i, env) \\ p_{\theta_3$	-	
$\begin{array}{lll} JS(\cdot \cdot) & \text{JS divergence} \\ q_{\phi_1}(\mathbf{z}_i \mathbf{h}_i,env) & \text{Distribution modeled by the NodeVAE encoder} \\ p_{\theta_1}(\mathbf{h}_i \mathbf{z}_i,env) & \text{Distribution modeled by the NodeVAE decoder} \\ p_{\theta_2}(\mathbf{z}_G G,env) & \text{Distribution modeled by the GVAG encoder} \\ p_{\theta_2}(\mathbf{v}_i \mathbf{z}_G,\mathbf{z}_i,env) & \text{Node existence probability distribution modeled by} \\ GVAG decoder & GVAG decoder \\ p(\mathbf{z}) & \text{Prob}(\mathbf{z}_i) & \text{Node existence probability of node } i \\ Prob(\mathbf{e}_{ij}) & \text{Edge existence probability of node } i \\ Prob(\mathbf{e}_{ij}) & \text{Edge existence probability of node } i \\ E_{RR} & \text{Reconstruction loss} MI \text{ loss} \\ \mathcal{L}_{NodeVAE} & \text{NodeVAE loss} \\ \mathcal{R}_{\text{causal}} & \text{Causal structure regularization} \\ \mathcal{R}_{\text{hinge}} & \text{Hinge regularization} \\ \mathcal{R}_{\text{subg,node}} & \text{Subgraph node count regularization} \\ \mathcal{L}_{CON} & \text{Contrastive loss} \\ \text{LAR} & \text{Last action rewards} \\ \text{fid}_{-} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_v & \text{Node density relative to the original graph} \\ \mathcal{L}_{\text{edge density relative to the original graph} \\ \mathcal{L}_{e$		
$\begin{array}{lll} q_{\phi_1}(\mathbf{z}_i \mathbf{h}_i,env) & \text{Distribution modeled by the NodeVAE encoder} \\ p_{\theta_i}(\mathbf{h}_i \mathbf{z}_i,env) & \text{Distribution modeled by the NodeVAE decoder} \\ p_{\theta_2}(\mathbf{z}_c G,env) & \text{Distribution modeled by the GVAG encoder} \\ p_{\theta_2}(\mathbf{v}_i \mathbf{z}_G,\mathbf{z}_i,env) & \text{Node existence probability distribution modeled by} \\ quad GVAG decoder & \text{Cover} \\ p(\mathbf{z}) & \text{Edge existence probability distribution modeled by} \\ QVAG decoder & p(\mathbf{z}) & \text{Edge existence probability of node } i \\ Prob(v_i) & \text{Edge existence probability of node } i \\ Prob(e_{ij}) & \text{Edge existence probability of edge } e_{ij} \\ \mathcal{L}_{\text{MI}} & \text{Subgraph reconstruction loss} MI \text{ loss} \\ \mathcal{L}_{\text{RR}} & \text{Reconstruction regularization loss} \\ \mathcal{L}_{\text{CodeVAE}} & \text{NodeVAE loss} \\ \mathcal{R}_{\text{causal}} & \text{Causal structure regularization} \\ \mathcal{R}_{\text{hinge}} & \text{Subgraph node count regularization} \\ \mathcal{L}_{\text{CON}} & \text{Contrastive loss} \\ \text{LAR} & \text{Last action rewards} \\ \text{fid}_{-} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_v & \text{Node density relative to the original graph} \\ \mathcal{L}_{\text{edge}} & \text{density relative to the original graph} \\ \text{Edge density relative to the original graph} \\ \text{Edge density relative to the original graph} \\ \text{Edge density relative to the original graph} \\ \end{array}$		
$\begin{array}{lll} q_{\phi_2}(\mathbf{z}_G G,env) & \text{Distribution modeled by the GVAG encoder} \\ p_{\theta_2}(v_i \mathbf{z}_G,\mathbf{z}_i,env) & \text{Node existence probability distribution modeled by} \\ GVAG decoder \\ p_{\theta_3}(e_{ij} \mathbf{z}_G,\mathbf{z}_i,\mathbf{z}_j,env) & \text{GVAG decoder} \\ p(\mathbf{z}) & \text{GVAG decoder} \\ prob(v_i) & \text{Node existence probability distribution modeled by} \\ Prob(v_i) & \text{Node existence probability of node } i \\ Prob(e_{ij}) & \text{Edge existence probability of node } i \\ L_{\text{MI}} & \text{Subgraph reconstruction loss} MI \text{ loss} \\ L_{\text{RR}} & \text{Reconstruction regularization loss} \\ \mathcal{L}_{\text{NodeVAE}} & \text{NodeVAE} \text{ loss} \\ \mathcal{R}_{\text{causal}} & \text{Causal structure regularization} \\ \mathcal{R}_{\text{hinge}} & \text{Hinge regularization} \\ \mathcal{R}_{\text{subg},\text{node}} & \text{Subgraph node count regularization} \\ \mathcal{L}_{\text{CON}} & \text{Contrastive loss} \\ \text{LAR} & \text{Last action rewards} \\ \text{fid}_{-} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_v & \text{Node density relative to the original graph} \\ \rho_e & \text{Edge density relative to the original graph} \\ \end{array}$		
$\begin{array}{lll} p_{\theta_2}(v_i \mathbf{z}_G,\mathbf{z}_i,env) & \text{Node existence probability distribution modeled by} \\ \text{GVAG decoder} \\ p_{\theta_3}(e_{ij} \mathbf{z}_G,\mathbf{z}_i,\mathbf{z}_j,env) & \text{Edge existence probability distribution modeled by} \\ \text{GVAG decoder} \\ p(\mathbf{z}) & \text{Frob}(v_i) & \text{Prob}(v_i) & \text{Node existence probability of node } i \\ \text{Prob}(e_{ij}) & \text{Edge existence probability of node } i \\ \text{Prob}(e_{ij}) & \text{Edge existence probability of node } i \\ \text{Em} & \text{Subgraph reconstruction loss} MI \text{ loss} \\ \mathcal{L}_{\text{RR}} & \text{Reconstruction regularization loss} \\ \mathcal{L}_{\text{NodeVAE}} & \text{NodeVAE loss} \\ \mathcal{R}_{\text{causal}} & \text{Causal structure regularization} \\ \mathcal{R}_{\text{hinge}} & \text{Hinge regularization} \\ \mathcal{R}_{\text{subg.node}} & \text{Subgraph node count regularization} \\ \mathcal{L}_{\text{CON}} & \text{Contrastive loss} \\ \text{LAR} & \text{Last action rewards} \\ \text{fid}_{-} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_v & \text{Node density relative to the original graph} \\ \rho_e & \text{Edge density relative to the original graph} \\ \end{array}$	$p_{\theta_1}(\mathbf{h}_i \mathbf{z}_i, env)$	Distribution modeled by the NodeVAE decoder
$\begin{array}{c} \operatorname{GVAG} \operatorname{decoder} \\ p_{\theta_3}(e_{ij} \mathbf{z}_G,\mathbf{z}_i,\mathbf{z}_j,env) \\ \end{array} \\ \begin{array}{c} \operatorname{Edge} \operatorname{existence} \operatorname{probability} \operatorname{distribution} \operatorname{modeled} \operatorname{by} \\ \operatorname{GVAG} \operatorname{decoder} \\ p(\mathbf{z}) \\ \operatorname{Prob}(v_i) \\ \operatorname{Prob}(e_{ij}) \\ \end{array} \\ \begin{array}{c} \operatorname{Prob}(v_i) \\ \operatorname{Prob}(e_{ij}) \\ \end{array} \\ \begin{array}{c} \operatorname{Edge} \operatorname{existence} \operatorname{probability} \operatorname{of} \operatorname{nod} e i \\ \operatorname{Prob}(e_{ij}) \\ \end{array} \\ \begin{array}{c} \operatorname{Edge} \operatorname{existence} \operatorname{probability} \operatorname{of} \operatorname{nod} e i \\ \end{array} \\ \operatorname{Edge} \operatorname{existence} \operatorname{probability} \operatorname{of} \operatorname{nod} e i \\ \end{array} \\ \begin{array}{c} \operatorname{Edge} \operatorname{existence} \operatorname{probability} \operatorname{of} \operatorname{nod} e i \\ \end{array} \\ \operatorname{Edge} \operatorname{existence} \operatorname{probability} \operatorname{of} \operatorname{ofde} e e_{ij} \\ \operatorname{Subgraph} \operatorname{reconstruction} \operatorname{loss}/MI \operatorname{loss} \\ \\ \mathcal{L}_{\operatorname{NodeVAE}} \operatorname{Inss} \\ \mathcal{R}_{\operatorname{causal}} \\ \mathcal{R}_{\operatorname{causal}} \\ \mathcal{R}_{\operatorname{causal}} \\ \mathcal{R}_{\operatorname{causal}} \\ \mathcal{R}_{\operatorname{subg-node}} \\ \mathcal{L}_{\operatorname{CON}} \\ \operatorname{Contrastive} \operatorname{loss} \\ \operatorname{LAR} \\ \operatorname{Last} \operatorname{action} \operatorname{rewards} \\ \operatorname{fid}_{\operatorname{log}} \\ \operatorname{fid}_{\operatorname{log}} \\ \operatorname{fid}_{\operatorname{log}} \\ \operatorname{Positive} \operatorname{Fidelity} \\ \operatorname{GEF} \\ \operatorname{Unfaithfulness} \\ \\ \rho_v \\ \operatorname{Node} \operatorname{density} \operatorname{relative} \operatorname{to} \operatorname{the} \operatorname{original} \operatorname{graph} \\ \operatorname{Edge} \operatorname{density} \operatorname{relative} \operatorname{to} \operatorname{the} \operatorname{original} \operatorname{graph} \\ \end{array} \\ \end{array}$	$q_{\phi_2}(\mathbf{z}_G G,env)$	
$\begin{array}{lll} p_{\theta_3}(e_{ij} \mathbf{z}_G,\mathbf{z}_i,\mathbf{z}_j,env) & \text{Edge existence probability distribution modeled by} \\ & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & \\ & & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & \\ & & & & \\ & & & \\ & & & & \\ & & $	$p_{\theta_2}(v_i \mathbf{z}_G,\mathbf{z}_i,env)$	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	(1	
$\begin{array}{lll} p(\mathbf{z}) & \text{Prior distribution of graph-invariant representations} \\ \text{Prob}(v_i) & \text{Node existence probability of node } i \\ \text{Prob}(e_{ij}) & \text{Edge existence probability of edge } e_{ij} \\ \mathcal{L}_{\text{MI}} & \text{Subgraph reconstruction loss} MI \text{ loss} \\ \mathcal{L}_{\text{RR}} & \text{Reconstruction regularization loss} \\ \mathcal{L}_{\text{NodeVAE}} & \text{Insection regularization loss} \\ \mathcal{R}_{\text{causal}} & \text{Causal structure regularization} \\ \mathcal{R}_{\text{hinge}} & \text{Hinge regularization} \\ \mathcal{R}_{\text{subg_node}} & \text{Subgraph node count regularization} \\ \mathcal{L}_{\text{CON}} & \text{Contrastive loss} \\ \text{LAR} & \text{Last action rewards} \\ \text{fid}_{-} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_v & \text{Node density relative to the original graph} \\ \rho_e & \text{Edge density relative to the original graph} \\ \end{array}$	$p_{\theta_3}(e_{ij} \mathbf{z}_G,\mathbf{z}_i,\mathbf{z}_j,env)$	
$\begin{array}{lll} \operatorname{Prob}(v_i) & \operatorname{Node} \ \operatorname{existence} \ \operatorname{probability} \ \operatorname{ofnode} \ i \\ \operatorname{Prob}(v_i) & \operatorname{Edge} \ \operatorname{existence} \ \operatorname{probability} \ \operatorname{ofedge} \ e_{ij} \\ \mathcal{L}_{\operatorname{MI}} & \operatorname{Subgraph} \ \operatorname{reconstruction} \ \operatorname{loss} \ \mathcal{M} \ \operatorname{loss} \\ \mathcal{L}_{\operatorname{RR}} & \operatorname{Reconstruction} \ \operatorname{regularization} \ \operatorname{loss} \\ \mathcal{L}_{\operatorname{NodeVAE}} & \operatorname{NodeVAE} \ \operatorname{loss} \\ \mathcal{R}_{\operatorname{causal}} & \operatorname{Causal structure} \ \operatorname{regularization} \\ \mathcal{R}_{\operatorname{hinge}} & \operatorname{Hinge} \ \operatorname{regularization} \\ \mathcal{R}_{\operatorname{subg,node}} & \operatorname{Subgraph} \ \operatorname{node} \ \operatorname{count} \ \operatorname{regularization} \\ \mathcal{L}_{\operatorname{CON}} & \operatorname{Contrastive} \ \operatorname{loss} \\ \operatorname{LAR} & \operatorname{Last} \ \operatorname{action} \ \operatorname{rewards} \\ \operatorname{fid}_{-} & \operatorname{Negative} \ \operatorname{Fidelity} \\ \operatorname{fid}_{+} & \operatorname{Positive} \ \operatorname{Fidelity} \\ \operatorname{GEF} & \operatorname{Unfaithfulness} \\ \rho_v & \operatorname{Node} \ \operatorname{density} \ \operatorname{relative} \ \operatorname{to} \ \operatorname{the} \ \operatorname{original} \ \operatorname{graph} \\ \rho_e & \operatorname{Edge} \ \operatorname{density} \ \operatorname{relative} \ \operatorname{to} \ \operatorname{the} \ \operatorname{original} \ \operatorname{graph} \\ \end{array}$	$n(\mathbf{z})$	
$\begin{array}{lll} \operatorname{Prob}(e_{ij}) & \operatorname{Edge} \ \operatorname{existence} \ \operatorname{probability} \ \operatorname{of} \ \operatorname{edge} \ e_{ij} \\ \mathcal{L}_{\operatorname{MI}} & \operatorname{Subgraph} \ \operatorname{reconstruction} \ \operatorname{loss} MI \ \operatorname{loss} \\ \mathcal{L}_{\operatorname{RR}} & \operatorname{Reconstruction} \ \operatorname{regularization} \ \operatorname{loss} \\ \mathcal{L}_{\operatorname{NodeVAE}} & \operatorname{NodeVAE} \ \operatorname{loss} \\ \mathcal{R}_{\operatorname{causal}} & \operatorname{Causal} \ \operatorname{structure} \ \operatorname{regularization} \\ \mathcal{R}_{\operatorname{hinge}} & \operatorname{Hinge} \ \operatorname{regularization} \\ \mathcal{R}_{\operatorname{Subg_node}} & \operatorname{Subgraph} \ \operatorname{node} \ \operatorname{count} \ \operatorname{regularization} \\ \mathcal{L}_{\operatorname{CON}} & \operatorname{Contrastive} \ \operatorname{loss} \\ \operatorname{LAR} & \operatorname{Last} \ \operatorname{action} \ \operatorname{rewards} \\ \operatorname{fid}_{-} & \operatorname{Negative} \ \operatorname{Fidelity} \\ \operatorname{fid}_{+} & \operatorname{Positive} \ \operatorname{Fidelity} \\ \operatorname{GEF} & \operatorname{Unfaithfulness} \\ \rho_v & \operatorname{Node} \ \operatorname{density} \ \operatorname{relative} \ \operatorname{to} \ \operatorname{the} \ \operatorname{original} \ \operatorname{graph} \\ \rho_e & \operatorname{Edge} \ \operatorname{density} \ \operatorname{relative} \ \operatorname{to} \ \operatorname{the} \ \operatorname{original} \ \operatorname{graph} \\ \end{array}$		
$ \begin{array}{lll} \mathcal{L}_{\text{MII}} & \text{Subgraph reconstruction loss}/M\overline{I} \text{loss} \\ \mathcal{L}_{\text{RR}} & \text{Reconstruction regularization loss} \\ \mathcal{L}_{\text{NodeVAE}} & \text{NodeVAE loss} \\ \mathcal{R}_{\text{caussal}} & \text{Causal structure regularization} \\ \mathcal{R}_{\text{hinge}} & \text{Hinge regularization} \\ \mathcal{R}_{\text{subg_node}} & \text{Subgraph node count regularization} \\ \mathcal{L}_{\text{CON}} & \text{Contrastive loss} \\ \text{LAR} & \text{Last action rewards} \\ \text{fid}_{-} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_{v} & \text{Node density relative to the original graph} \\ \rho_{e} & \text{Edge density relative to the original graph} \\ \end{array} $		
$ \begin{array}{cccc} \mathcal{L}_{RR} & Reconstruction regularization loss \\ \mathcal{L}_{NodeVAE} & NodeVAE loss \\ \mathcal{R}_{causal} & Causal structure regularization \\ \mathcal{R}_{hinge} & Hinge regularization \\ \mathcal{R}_{subg,node} & Subgraph node count regularization \\ \mathcal{L}_{CON} & Contrastive loss \\ LAR & Last action rewards \\ fid_{-} & Negative Fidelity \\ fid_{+} & Positive Fidelity \\ GEF & Unfaithfulness \\ \rho_{v} & Node density relative to the original graph \\ \rho_{e} & Edge density relative to the original graph \\ \end{array} $		
$ \begin{array}{lll} \mathcal{R}_{\text{causal}} & \text{Causal structure regularization} \\ \mathcal{R}_{\text{hinge}} & \text{Hinge regularization} \\ \mathcal{R}_{\text{subg_node}} & \text{Subgraph node count regularization} \\ \mathcal{L}_{\text{CON}} & \text{Contrastive loss} \\ \text{LAR} & \text{Last action rewards} \\ \text{fid}_{-} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_{v} & \text{Node density relative to the original graph} \\ \rho_{e} & \text{Edge density relative to the original graph} \\ \end{array} $		Reconstruction regularization loss
$ \begin{array}{lll} \mathcal{R}_{\text{hinge}} & \text{Hinge regularization} \\ \mathcal{R}_{\text{subg_node}} & \text{Subgraph node count regularization} \\ \mathcal{L}_{\text{CON}} & \text{Contrastive loss} \\ \text{LAR} & \text{Last action rewards} \\ \text{fid}_{-} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_{v} & \text{Node density relative to the original graph} \\ \rho_{e} & \text{Edge density relative to the original graph} \\ \end{array} $	$\mathcal{L}_{ ext{NodeVAE}}$	
$ \begin{array}{lll} \mathcal{R}_{\text{subg_node}} & \text{Subgraph node count regularization} \\ \mathcal{L}_{\text{CON}} & \text{Contrastive loss} \\ \text{LAR} & \text{Last action rewards} \\ \text{fid}_{-} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_{v} & \text{Node density relative to the original graph} \\ \rho_{e} & \text{Edge density relative to the original graph} \\ \end{array} $		Causal structure regularization
	$\mathcal{R}_{\mathrm{hinge}}$	
LAR Last action rewards fid Negative Fidelity fid_+ Positive Fidelity GEF Unfaithfulness ρ_v Node density relative to the original graph ρ_e Edge density relative to the original graph		
$ \begin{array}{lll} \text{fid}_{-} & \text{Negative Fidelity} \\ \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_{v} & \text{Node density relative to the original graph} \\ \rho_{e} & \text{Edge density relative to the original graph} \\ \end{array} $		
$\begin{array}{ll} \text{fid}_{+} & \text{Positive Fidelity} \\ \text{GEF} & \text{Unfaithfulness} \\ \rho_{v} & \text{Node density relative to the original graph} \\ \rho_{e} & \text{Edge density relative to the original graph} \end{array}$		
$ \begin{array}{ll} \text{GEF} & \text{Unfaithfulness}' \\ \rho_v & \text{Node density relative to the original graph} \\ \rho_e & \text{Edge density relative to the original graph} \\ \end{array} $		
$ ho_v$ Node density relative to the original graph $ ho_e$ Edge density relative to the original graph		
ρ_e Edge density relative to the original graph		
	$ ho_e$	
	T	Response time (second)

Table 6: List of symbols and their descriptions.

A.2 Subgraph Reconstruction Algorithm

Algorithm 1 presents the pseudocode for generating subgraphs during the training phase. In essence, during the training phase, we first sample nodes that will appear on the explanation subgraph based on their existence probability, assigning these sampled nodes a probability of 1.0. Then, we recalculate the existence probabilities for each edge by integrating

Algorithm 1 Sample-based subgraph reconstruction algorithm

```
Require: Input parameters:
          edge_index: Edge set of original graph
          node_prob: Node existence probability
          link prob: Edge existence probability
          max_nodes: Maximum number of nodes in subgraph
          start_nid: Start node of generating subgraph
          density: Subgraph density limit
          max_iter: Maximum number of iterations
          min_edges: Minimum number of edges in subgraph
 1: Initialize:
 2: Ensure max_nodes does not exceed the number of nodes available
    in the original graph
 3: Adjust min_edges based on the density
 4: Add a small epsilon to node_prob and link_prob to prevent com-
    putational issues
 5: Initialize node selection vector current_node
    Sampling nodes for the subgraph:
 6: if start_nid is provided then
       Set the corresponding index in current_node to True
 8: end if
 9: Copy node\_prob as sampling probabilities Prob_n
10: for iter from 1 to max_iter do
        For nodes in current\_node, set Prob_n to -1
11:
        Sample new nodes based on Prob_n and update current\_node
12:
       if sum(current\_node) \ge max\_nodes then
13:
14:
15:
        end if
16: end for
17: if sum(current_node) < max_nodes then
18.
        For nodes in current node, set Prob_n to -1
        Select new nodes based on Prob_n and update current node
19:
20: else
21:
       Prune nodes from current_node based on node_prob to fit
    within max_nodes
22: end if
23: Reset Prob_n to node\_prob
24: For nodes in current_node, set Prob_n to 1
    Sampling edges for the subgraph based on the nodes within
    the subgraph:
25: Initialize edge selection vector current_edge
26: for iter from 1 to max_iter do
27:
        Copy link\_prob as sampling probabilities Prob_e
28:
        For links in current_edge, set Prob_e to 0
29:
        Recompute Prob_e using Prob_n and Prob_e
30:
        Sample links based on Prob_e
31:
        Update current_link based on sampled links
       if edge density > density then
32:
33:
           break
34:
        end if
35: end for
```

36: If node has no edge, remove it from *current_node* 37: Calculate total graph probability total_graph_prob based on

node_prob and link_prob

38: **return** *current_node*, *current_link*, *total_graph_prob*

Algorithm 2 Edge first reconstruction algorithm

Require: Input parameters:

edge_index: Edge set of original graph
node_prob: Node existence probability
link_prob: Edge existence probability

max_nodes: Maximum number of nodes in subgraph

start_nid: Start node of generating subgraph

density: Subgraph density limit

min_edges: Minimum number of edges in subgraph

Initialize:

- 1: Initialize node selection vector *current_node*
- 2: Initialize edge selection vector *current_edge*
- 3: Adjust min_edges based on density
- 4: Add a small epsilon to *node_prob* and *link_prob* to prevent computational issues

Prepare edge existence probability:

- 5: **if** *start_nid* is provided **then**
- 6: Set the corresponding index in *current_node* to True
- 7: **end if**
- 8: $total_graph_prob \leftarrow 0.0$
- 9: $current_node_prob \leftarrow node_prob$
- 10: $current_link_prob \leftarrow link_prob$
- Recompute current_link_prob using current_node_prob and current_link_prob
- 12: $max_edges \leftarrow ceil(density * |edge_index|)$
- 13: **if** $max_edges \le min_edges$ **then**
- 14: $max_edges \leftarrow min_edges$
- 15: end if

Select edges for the explanatory subgraph based on their probabilities:

- 16: sorted_edge_prob, sorted_eid ← topk(current_link_prob, k=max_edges)
- 17: For edge id in sorted_eid, set current_edge to True
- Calculate total graph probability total_graph_prob based on sorted_edge_prob

Update Nodes Based on Selected Edges:

- 19: Get source nodes of edges in current_edge, as src_nodes
- 20: Get destination nodes of edges in *current_edge*, as *dst_nodes*
- 21: For nodes in src_nodes or dst_nodes, set current_node to True
- 22: **return** current node, current link, total graph prob

both the node existence probability and the edge existence probability relevant to the current subgraph. Next, these recalculated probabilities are subsequently utilized to sample edges that will appear on the explanation subgraph. This methodology ensures that the probabilities of both nodes and edges are considered concurrently in generating the explanation subgraph, which can help in maintaining connectivity within the subgraph.

Moreover, through this random sampling process, GVAG effectively explores the entire space of potential subgraphs and avoids focusing on a limited set of nodes or edges, thereby enhancing the robustness and generalization of the generated explanation.

In the testing phase, GVAG directly uses the node existence probability and edge existence probability to calculate the final probability of each edge appearing on the explanation subgraph, and add these edges to the explanation subgraph according to the probability. The corresponding pseudocode is presented in Algorithm 2.

A.3 Computational Complexity Analysis of Subgraph Generation During Training Phase

We adopt a sampling-based subgraph generation algorithm, detailed in Algorithm 1, with its complexity influenced by various operations. The initialization steps, including adjustments to max_nodes and min_edges, are constant operations with a complexity of O(1), while adding ϵ to zero probabilities involves linear operations, resulting in a combined complexity of O(n) + O(m). The node sampling loop iterates up to max_iter times, with each iteration involving probability calculations and updates for all nodes, leading to a complexity of $O(n \cdot \text{max_iter})$. Similarly, the edge sampling loop processes all edges within each iteration, requiring recomputation of probabilities and sampling, contributing a complexity of $O(m \cdot \text{max iter})$. Post-processing adjustments, such as sorting and selecting top k elements for nodes and edges, add logarithmic factors, typically $O(n \log n)$ and $O(m \log m)$, respectively. Considering n > m, max_iter $> \log n$, and max_iter $> \log m$, the overall time complexity of the algorithm is $O(n \cdot \text{max_iter})$.

A.4 Computation Complexity Analysis of Subgraph Generation During Evaluation Phase

The computational complexity of the Algorithm 2 primarily depends on the operations performed on nodes and edges within the graph. Initially, adjusting the probability vectors for zero probabilities, which are operations linear in terms of the number of nodes n and edges m, contributes a complexity of O(n+m). This setup is followed by the critical step of selecting edges based on updated probabilities, involving a sorting operation. Since the edges are sorted to select the top edges based on their probability, this step incurs a complexity of $O(m \log m)$, which is the most computationally intensive part of the function. Updating the node selection based on the edges selected is relatively straightforward and operates linearly with respect to the number of selected edges, hence contributing an additional linear term. Overall, the sorting of edge probabilities dominates the computational complexity, making the function's total complexity mainly governed by $O(m \log m)$ with an additional linear component due to initialization and node updates based on selected edges.

B Supplementary Material on Experiments

B.1 Experimental Setup

Our experimental were conducted on an AMD EPYC 9754 CPU, an NVIDIA 4090D GPU with 24GB G6X memory, and 60GB of RAM. The software environment includes Python 3.10, CUDA 12.1, and PyTorch 2.1.0.

The Hyper-Parameter Settings. The key hyper-parameter settings for training and evaluation are shown in Table 7. These hyper-parameters are carefully selected based on prior empirical results and tuning experiments to balance model fidelity and efficiency.

B.2 Analysis of Hyper-Parameter Sensitivity

We also test the impact of several hyper-parameters on the quality of the final generated explanation subgraph, including

Hyper-parameters	Cora	Motif
Learning Rate	0.01	0.005
Weight Decay	1.00E-04	1.00E-04
Structure Infer Epochs	5	5
Number Environments K	4	5
Number Epochs	1	10
Batch Size	64	64
Prior Subgraph Max Nodes	60	7
Prior Subgraph Min Nodes	15	5
Prior Subgraph Density	0.35	0.1
Recon Loss Weight $\omega_{\rm RECON}$	2	2
Contrastive Loss Weight ω_{CON}	0.5	0.5
Last Action Rewards Weight ω_{LAR}	1	1

Table 7: Hyper-parameters settings.

edge density, last action rewards weight and reconstruction weight.

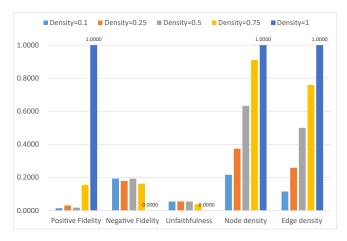


Figure 3: Hyper-parameter sensitivity study on different edge densities.

Different Edge Density. This study examines how edge density impacts the model's ability to generate effective explanations, with results summarized in Figure 3. As edge density increases from 0.1 to 1.0, a distinct trend in the performance metrics emerges:

- Positive Fidelity: Variations in positive fidelity suggest that explanation subgraphs with higher densities include more critical features essential for GNN predictions, thereby enhancing positive fidelity by better aligning with GNN predictions.
- Negative Fidelity: Adjustments in negative fidelity with varying densities indicate that lower densities, which result in sparser subgraphs, may omit crucial features necessary for the GNN's decision-making process.
- Unfaithfulness: Changes in unfaithfulness show that denser explanation subgraphs are likely to exhibit lower unfaithfulness, implying that denser subgraphs may better align with the predictions of the original graph, thus offering more faithful interpretations.

This analysis underscores the importance of managing edge density to balance the trade-offs between comprehensiveness

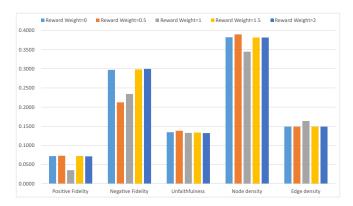


Figure 4: Hyper-parameter sensitivity study on different weights of LAR.

and simplicity in explanation subgraphs. Lower densities, while easier to interpret, might miss critical information; conversely, higher densities, though potentially more complex, provide a more detailed and accurate representation of the factors influencing the model's decisions. Achieving this balance is crucial for ensuring that explanations are both informative and practically useful, therefore meeting the needs of real-world applications.

Different Weights of Last Action Rewards. This study explores how varying the weights of the last action rewards influences OPEN's performance, with results presented in Figure 4.

- Positive Fidelity: Positive fidelity remains relatively stable across different reward weights, suggesting that adjustments in the last action rewards do not significantly affect the model's capability to include critical graph structures in the explanation subgraphs.
- Negative Fidelity: We observe a regular increase in negative fidelity as the reward weight increases, indicating that excessively high reward weights might overly penalize the model's errors, potentially leading to the exclusion of relevant structures.
- **Unfaithfulness:** Unfaithfulness demonstrates a decreasing trend with higher reward weights, which implies that the explanations become more aligned with the original graph's predictions, enhancing their faithfulness and reliability.

These findings indicate that while a higher reward weight can enhance the faithfulness of explanations, it may simultaneously compromise negative fidelity by penalizing the model too harshly. Therefore, it is essential to finely tune the last action rewards to maintain a balance between the depth and accuracy of the explanations produced by OPEN, ensuring that they are comprehensive yet precise.

Different Weights of Reconstruction Loss. This study examines the effect of varying reconstruction loss weights on OPEN's performance, with results illustrated in Figure 5.

• **Positive Fidelity:** Positive fidelity shows a positive correlation with increasing reconstruction loss weight, climbing from 0 to 2. This trend suggests that higher weights prompt

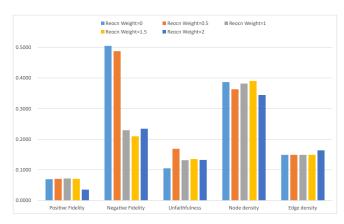


Figure 5: Hyper-parameter sensitivity study on different weights of reconstruction loss.

the model to preserve more crucial structures within the explanation subgraphs, thereby boosting positive fidelity.

- Negative Fidelity: Conversely, negative fidelity decreases as the reconstruction loss weight increases, highlighting that enhanced penalties for incorrect explanations help the model omit non-essential structures, thus refining the fidelity of its explanations.
- Unfaithfulness: Unfaithfulness does not exhibit a consistent trend with changes in reconstruction loss weight, indicating that while reconstruction loss aids in refining explanations, its effect on the faithfulness may be influenced by other factors within the model.

These findings illustrate the critical role that reconstruction loss weight plays in explanations generated by OPEN. Properly setting this weight can enhances the fidelity and accuracy of explanations.

B.3 Explanation Subgraph Examples

Figure 6 displays explanation subgraphs generated by the proposed OPEN and some other XGNN methods on the Motif dataset in the basis domain. The figure reveals that all methods, except PGMExplainer, successfully identify nodes relevant to predictions (blue nodes). OPEN stands out by considering both node and edge existence probabilities during subgraph generation, ensuring the connectivity of the explanation subgraphs. Moreover, OPEN's ability to adjust the size and density of the subgraphs based on prior knowledge offers a more flexible and user-friendly explanation approach compared to existing XGNN methods, enhancing user comprehension.

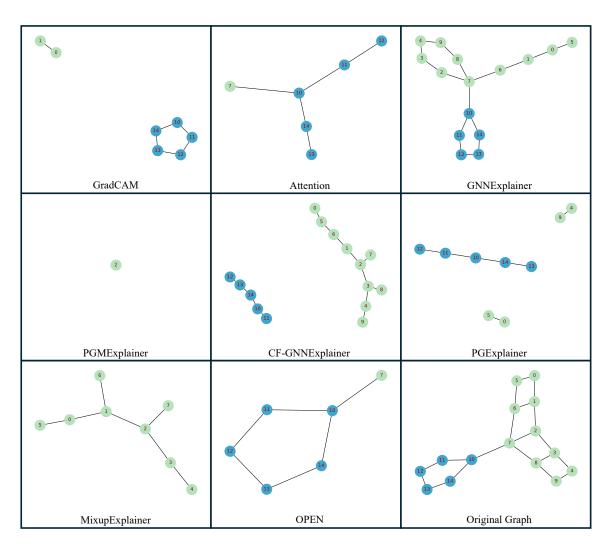


Figure 6: Explanation subgraphs.